

# The Complete PCI Express Reference Overview of Tutorial and Book

## Design Tools for PCI Express

The PCI Express specification is written as a specification. Thus, it is not organized by clear design topics, does not provide sufficient design details to easily master of PCI Express, and does not provide in depth illustrations and flowcharts to assist in designing components.

Intel recognized that simply summarizing or re-wording the specification as typically done in other design books by non-designers is insufficient. More extensive information by engineers for engineers is needed.

Intel also recognized that design and implementation information is of the greatest value when provided as part of focused design tools. Focused design tools save engineers weeks of trying to unravel the specification and provide the assurance of correct design implementation the first time.

To provide engineers with PCI Express design and implementation information that is easy and quick to master and detailed enough to assist in correct design, two of the many focused design tools available from Intel are as follows:

### **Six Detailed Tutorials**

**A new and exhaustively detailed design Book: *The Complete PCI Express Reference*.**

## Design Tools for PCI Express

Six free **Detailed Tutorials** ... One self paced tutorial for each of the Six Topic Groups. Each introduces PCI Express information with a narrative that complements detailed figures, flow charts, and tables for each specific Topic Group from the **Book**. The six free **Detailed Tutorials** are available at [www.intel.com/intelpress/pciexpresscomplete](http://www.intel.com/intelpress/pciexpresscomplete).

The primary focus design tool is *The Complete PCI Express Reference book* written by Edward Solari and Brad Congdon and published by Intel ... referred to as the “**Book**”.

The **Book** provides a complete and extensive narrative with detailed figures (over 250), detailed design flow charts, and exhaustive tables for the complete understanding and design assistance. The Book is over 1000 pages and can be ordered at [www.amazon.com](http://www.amazon.com) ... ISBN # 0971786194.

The purpose of this Tutorial and Book Overview is to introduce engineers to the extensive and detailed information available from these focused design tools but simply NOT available from the specification or any other design resource.

## PCI Express in Six Topic Groups

### Topic Group 1

**Tutgroup1: *Platform Architecture and Accessing of Resources within Architecture***

**References in the Book: *Chapters 1 to 4***

### Topic Group 2

**Tutgroup2 : *Packets' and Layers' Specifics and Errors***

**References in the Book: *Chapters 5 to 9***

### Topic Group 3

**Tutgroup3 : *Transaction Ordering and Flow Control Part 1 and 2 Protocols***

**References in the Book: *Chapters 10 to 12***

### Topic Group 4

**Tutgroup4 : *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

**References in the Book: *Chapters 13 to 17***

### Topic Group 5

**Tutgroup5 : *Other Hardware Topics***

**References in the Book: *Chapters 18 to 21***

### Topic Group 6

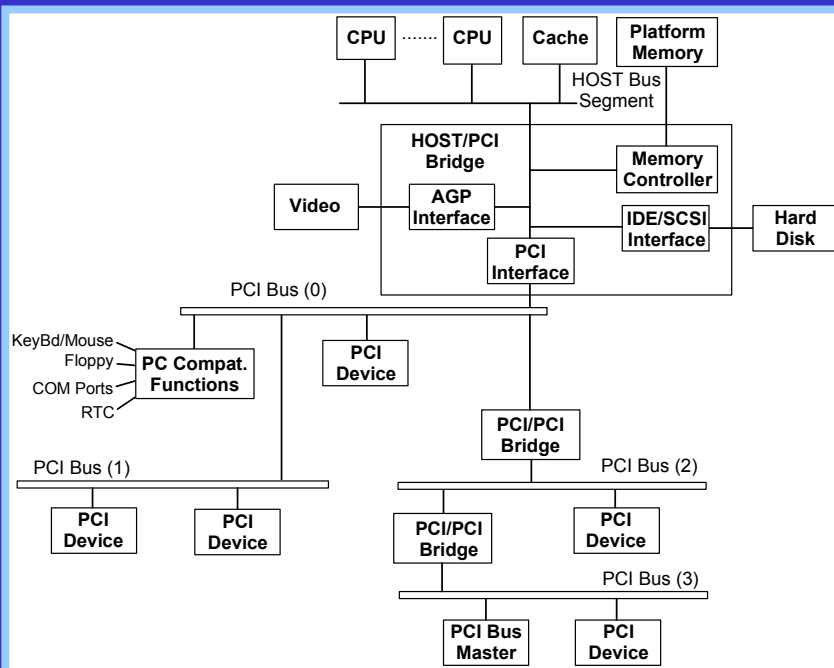
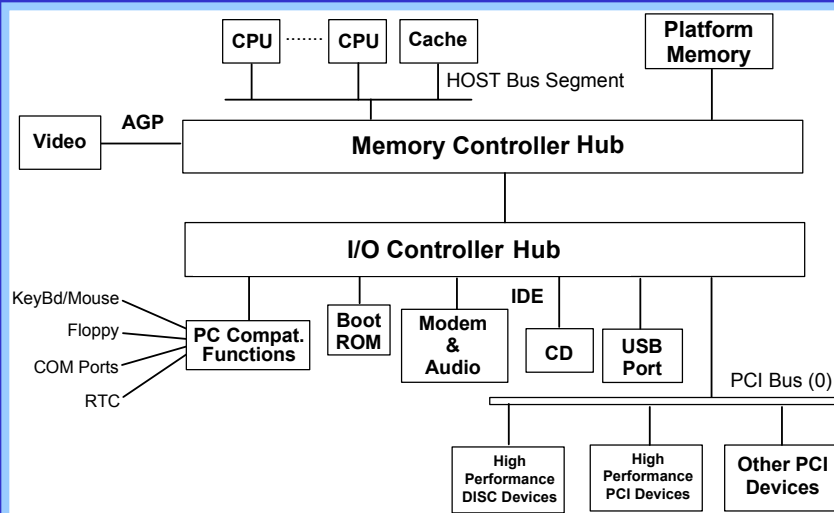
**Tutgroup6 : *Software Considerations***

**References in the Book: *Chapters 22 to 24***

## Sample Information from Design Tools for Topic Group 1

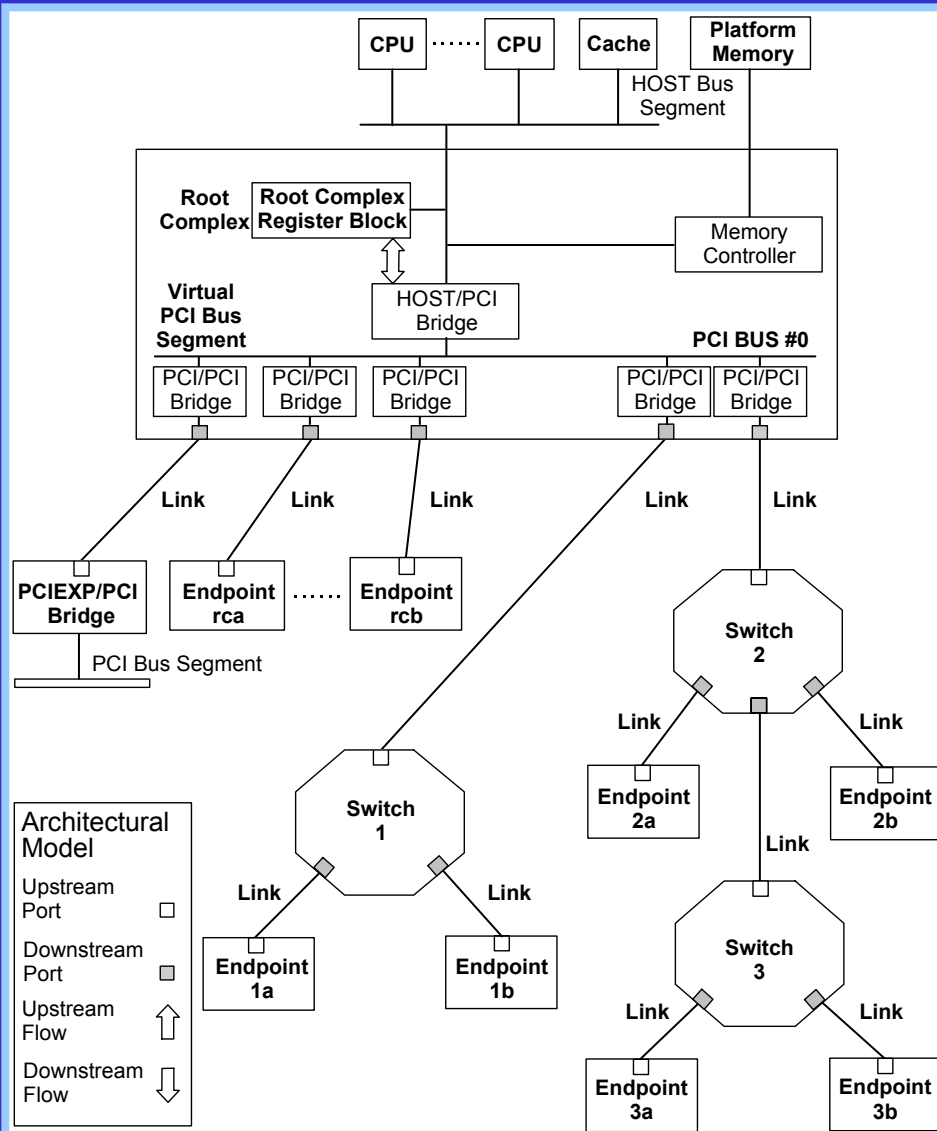
The PCI Express **platform architecture** is an evolution of PCI and PCI-X bus segments to a **point-to-point interconnections** called **links**, with retention of PCI elements for **software compatibility**.

Detailed Tutorial: *Platform Architecture and Accessing of Resources within Architecture*  
References in the Book: *Chapters 1 to 4*



## Sample Information for Topic Group 1

- The design tools provide **extensive discussion** with detailed features and tables that explain the evolution of PCI and PCI-X platforms from Memory and I/O Hubs to a HOST/PCI Bridge with a hierarchy of bus segments.



## Sample Information for Topic Group 1

- Discussions in the design tools **focus** on detailed block diagrams unavailable in the specifications.
- The **emphasis** of the block diagrams is the use of virtual PCI elements to retain compatibility with PCI software.

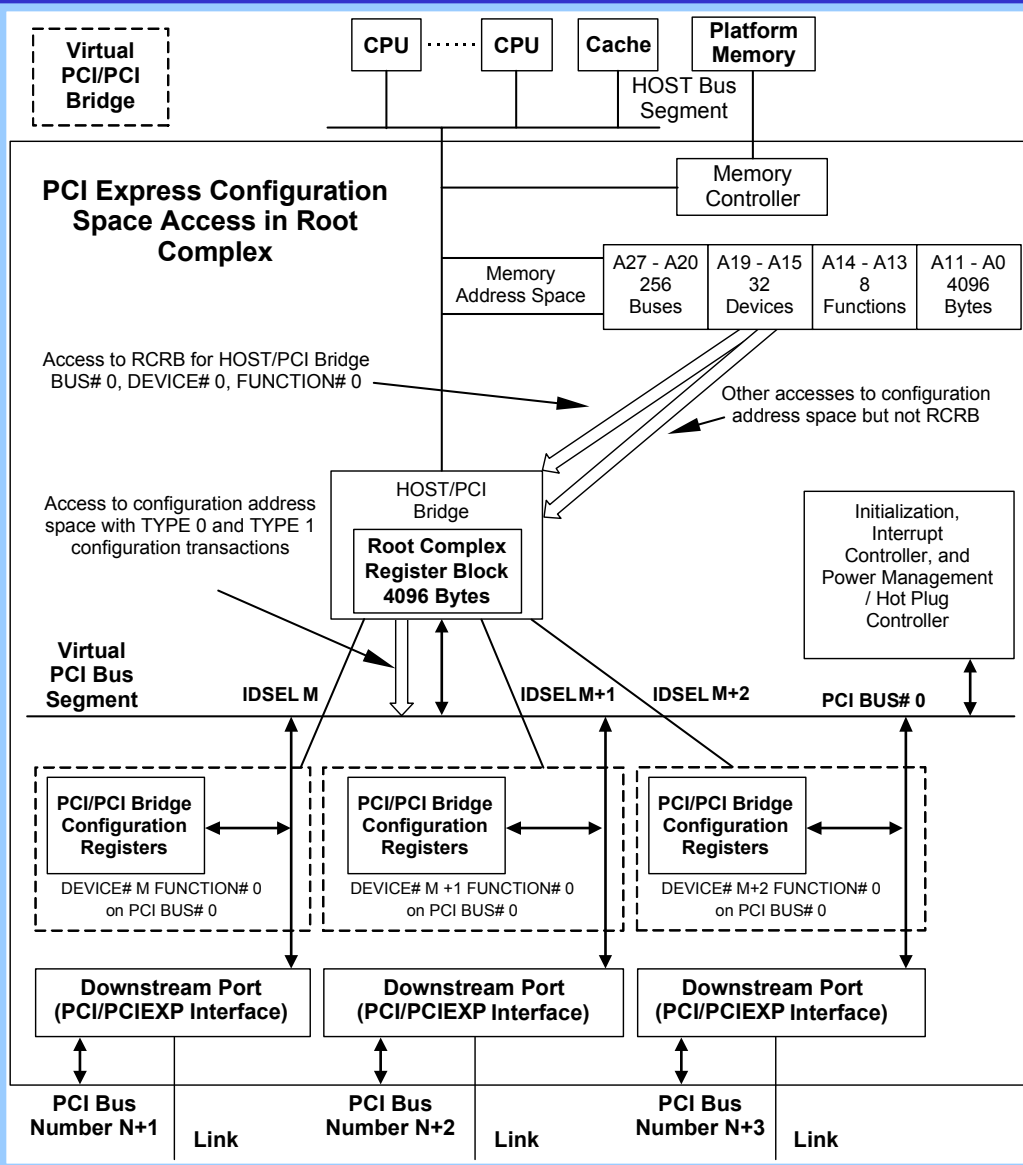
## Sample Information for Topic Group 1

- Extensive tables and narrative **explains** how PCI and PCI-X platforms' performance compares to PCI Express platforms' performance not available from the specification.

Data Bus Width	Signal Lines (excluding power, arbitration, and test)	Clock Signal Line Frequency (MHz)	Maximum Bandwidth Megabytes/Second	Maximum Add-in Card Slots
PCI-X 1.0 64 bits	82 lines	66.6	532.8	4 (1)
PCI-X 1.0 64 bits	82 lines	100	800	2 (1)
PCI-X 1.0 64 bits	82 lines	133.2	1065.6	1 (1)
PCI-X 2.0 64 bits	82 lines	Source synchronous DDR (3)	2131.2	1 (2)
PCI-X 2.0 64 bits	82 lines	Source synchronous QDR (3)	4262.4	1 (2)
PCI-X 3.0 64 bits	82 lines (4)	Source synchronous (3)	8524.8	1 (2)

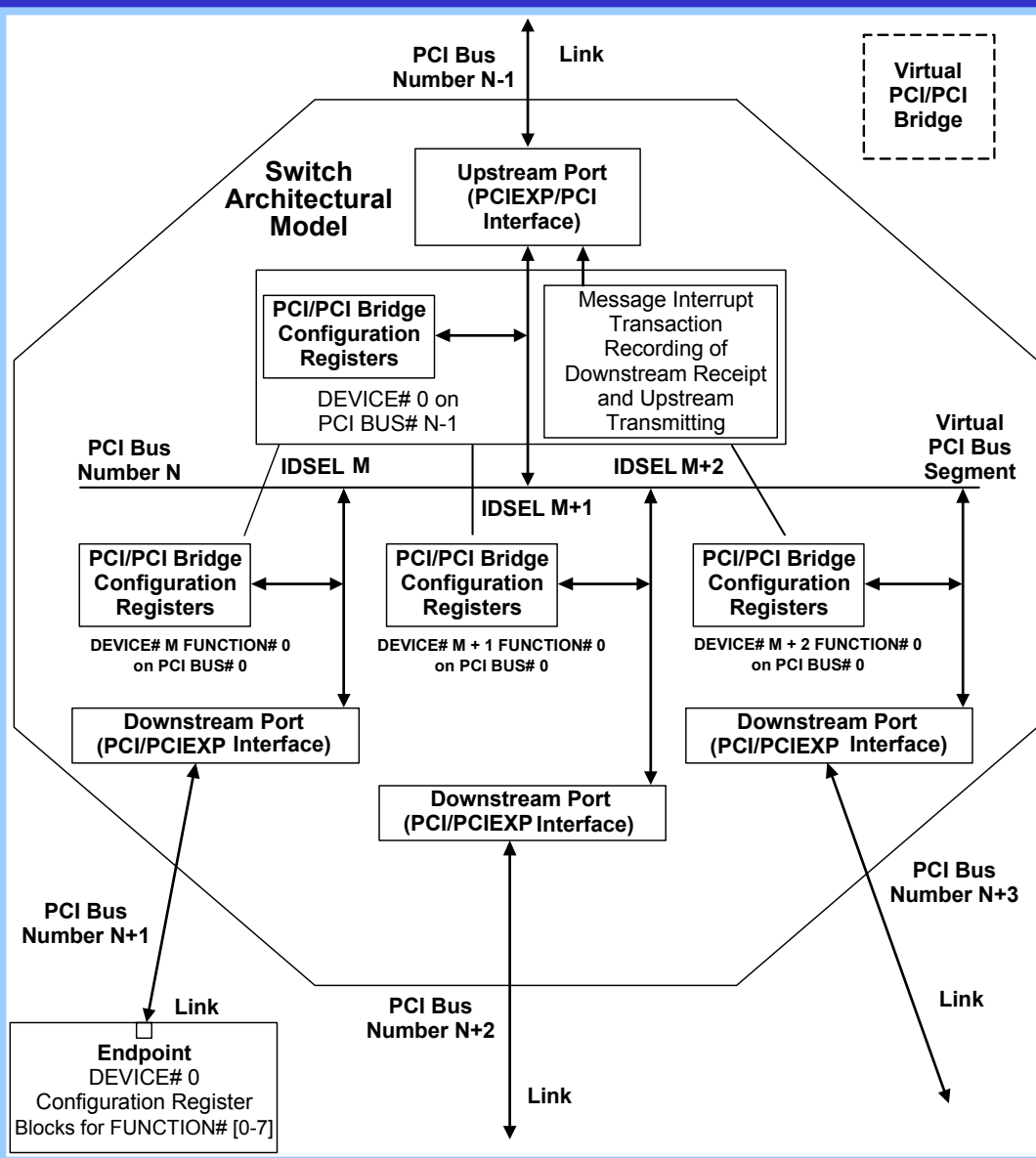
PCI Express Effective Bandwidth of Link Layer Transaction Packets per a Link 2.5Gb/sec (x1) with Raw 250 MB/s versus 80 Gb/sec (x32) with Raw 8000MB/s Per Differential Driven Pair in Each Direction not including bandwidth used by lane fillers, Ordered Sets, and Data Link Layer Packets						
Data Bytes Payload per Packet	128	256	512	1024	2048	4096
Efficiency						
32 Bits Address	84.21%	91.42%	95.52%	98.84%	98.84%	99.42%
64 Bits Address	83.12%	90.78%	95.17%	97.52%	98.75%	99.37%
Effective Data Bandwidth per Second for 32 Bits Address	210.5 MB/s versus 6736.8 MB/s	228.6 MB/s versus 7315.2 MB/s	237.5 MB/s versus 7600 MB/s	244.2 MB/s versus 7817.6 MB/s	247.1 MB/s versus 7907.2 MB/s	248.6 MB/s versus 7955.2 MB/s
Effective Data Bandwidth per Second for 64 Bits Address	207.5 MB/s versus 6640 MB/s	227.0 MB/s versus 7264 MB/s	237.9 MB/s versus 7613.6 MB/s	243.8 MB/s versus 7801.6 MB/s	246.9 MB/s versus 7900 MB/s	247.5 MB/s versus 7920 MB/s





## Sample Information for Topic Group 1

- All PCI Express devices are illustrated with the **greatest level of detail** to provide design insights not available from the specifications.
- An **example** is the Root Complex which replaces the HOST/PCI Bridge.

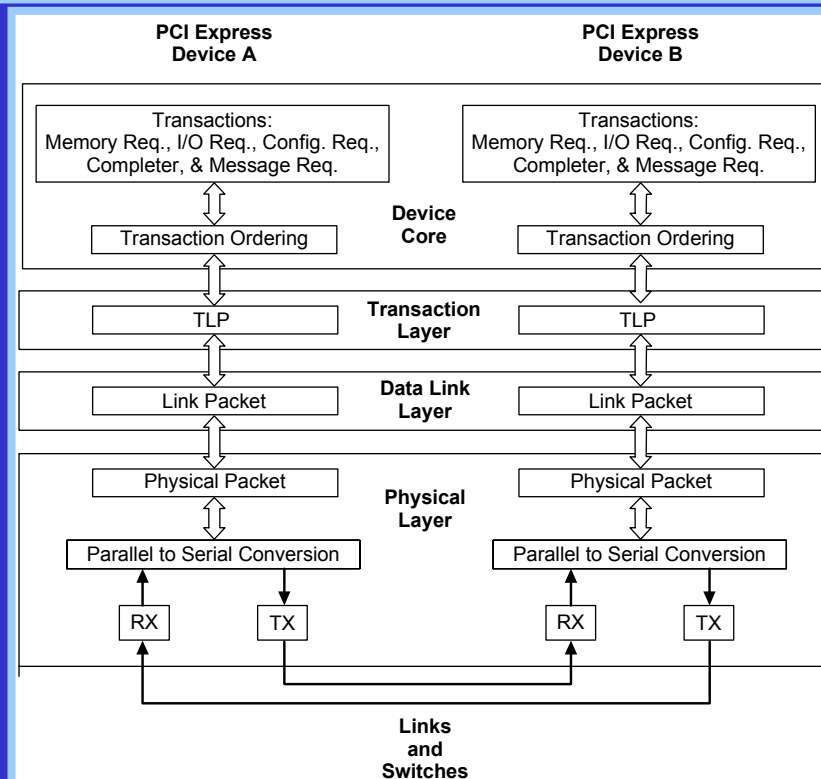
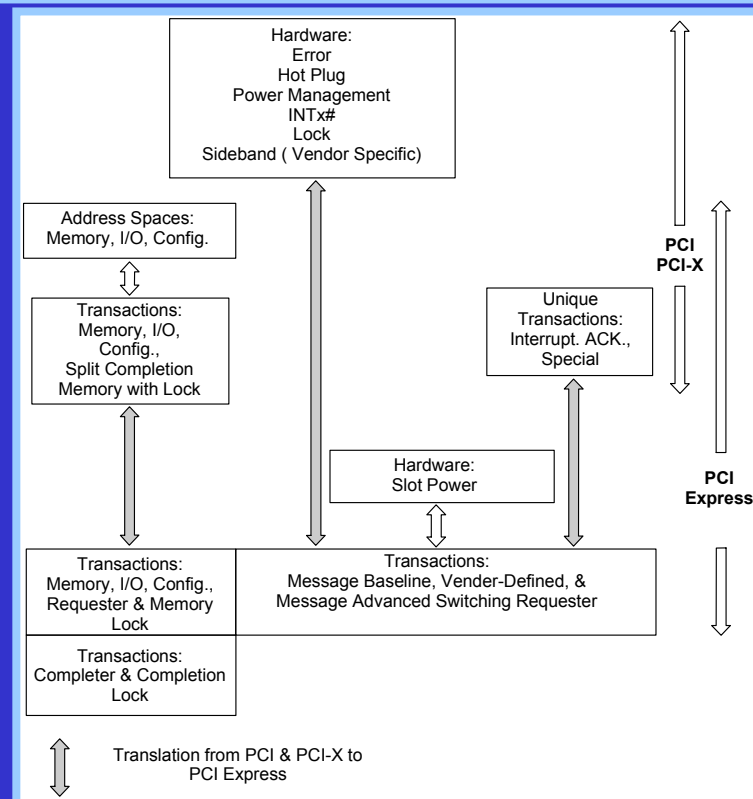


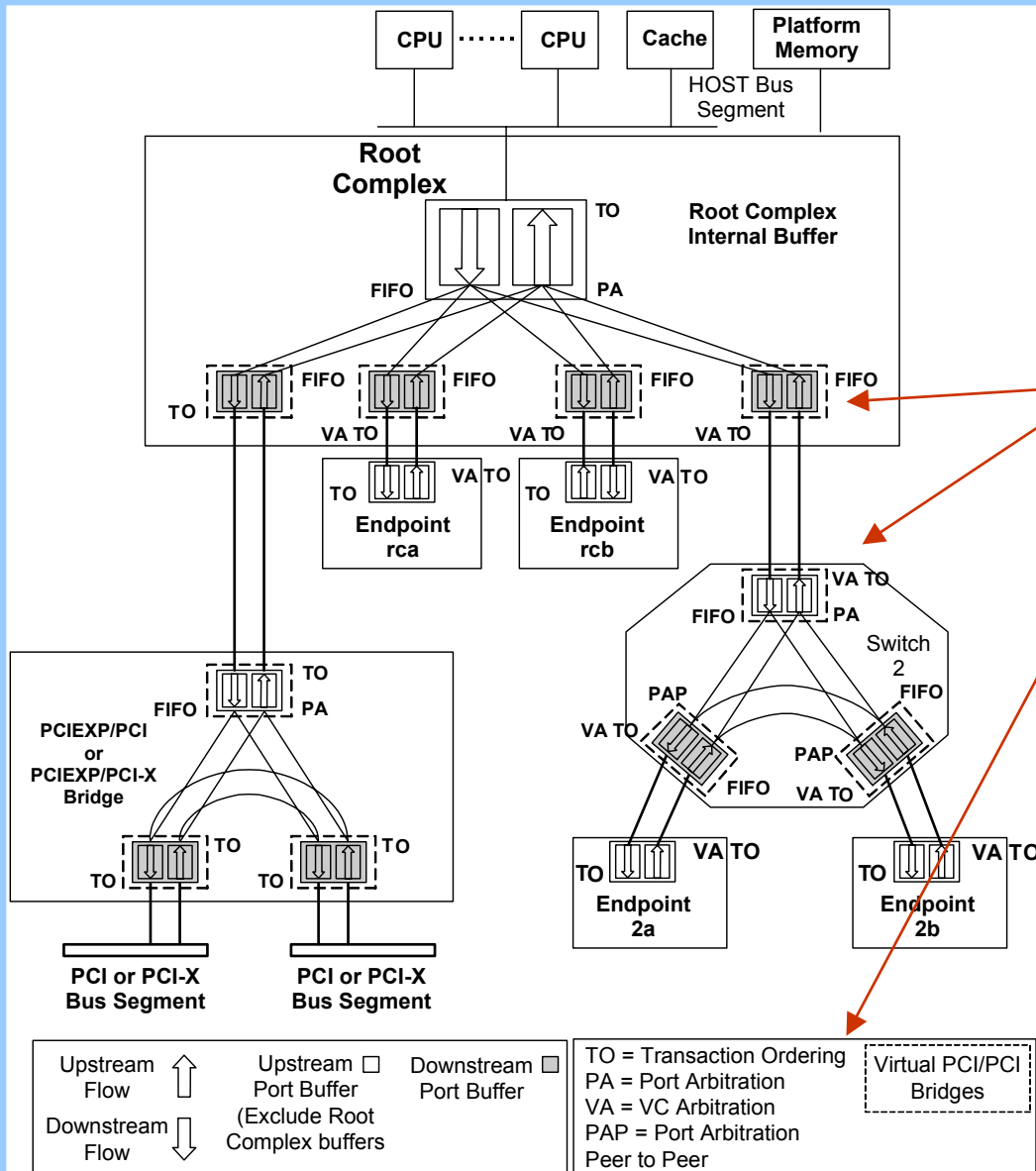
## Sample Information for Topic Group 1

- Another **example** of the detailed block diagrams unavailable in the specifications is the switch which emphasize point-to-point interconnection ... replaces the PCI/PCI Bridge which emphasize bus segments.
- Similar detailed block diagrams are developed in the design tools and discuss in detail for endpoints and bridges.

## Sample Information for Topic Group 1

- The design tools **clarify** how PCI Express has replaced both the PCI transactions and PCI hardware with only transactions to minimize the signal lines between devices.
- The interaction between PCI Express devices is via packets. The design tools go beyond the specifications by **detailing** how the packets are formed in multiple layers within each PCI Express device.





## Sample Information for Topic Group 1

- Unlike the specification which only vaguely references the buffers in the PCI Express platform, the design tools **detail** the buffers.
- The buffer details includes the architecture of buffers within all PCI Express devices.
- The flow of transactions throughout the PCI Express platform relies Requester/Completer protocol based on requester transactions and completer transactions. The design tools detail how the transactions flow **from buffer to buffer** using this protocol.

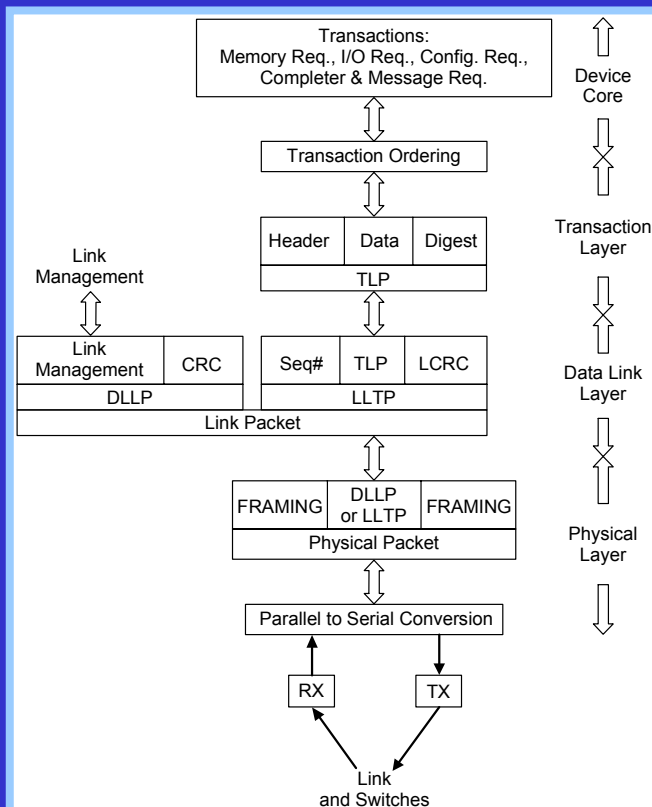
## Sample Information from Design Tools for Topic Group 2

The interaction between PCI Express device cores and the interconnecting links is done with transaction packets processed on **three layers** with each processing a unique **packet**. The conversion and transmission of packets may incur **errors** that must be checked and reported.

**Detailed Tutorial: *Packets' and Layers' Specifics and Errors***  
**References in the Book: *Chapters 5 to 9***

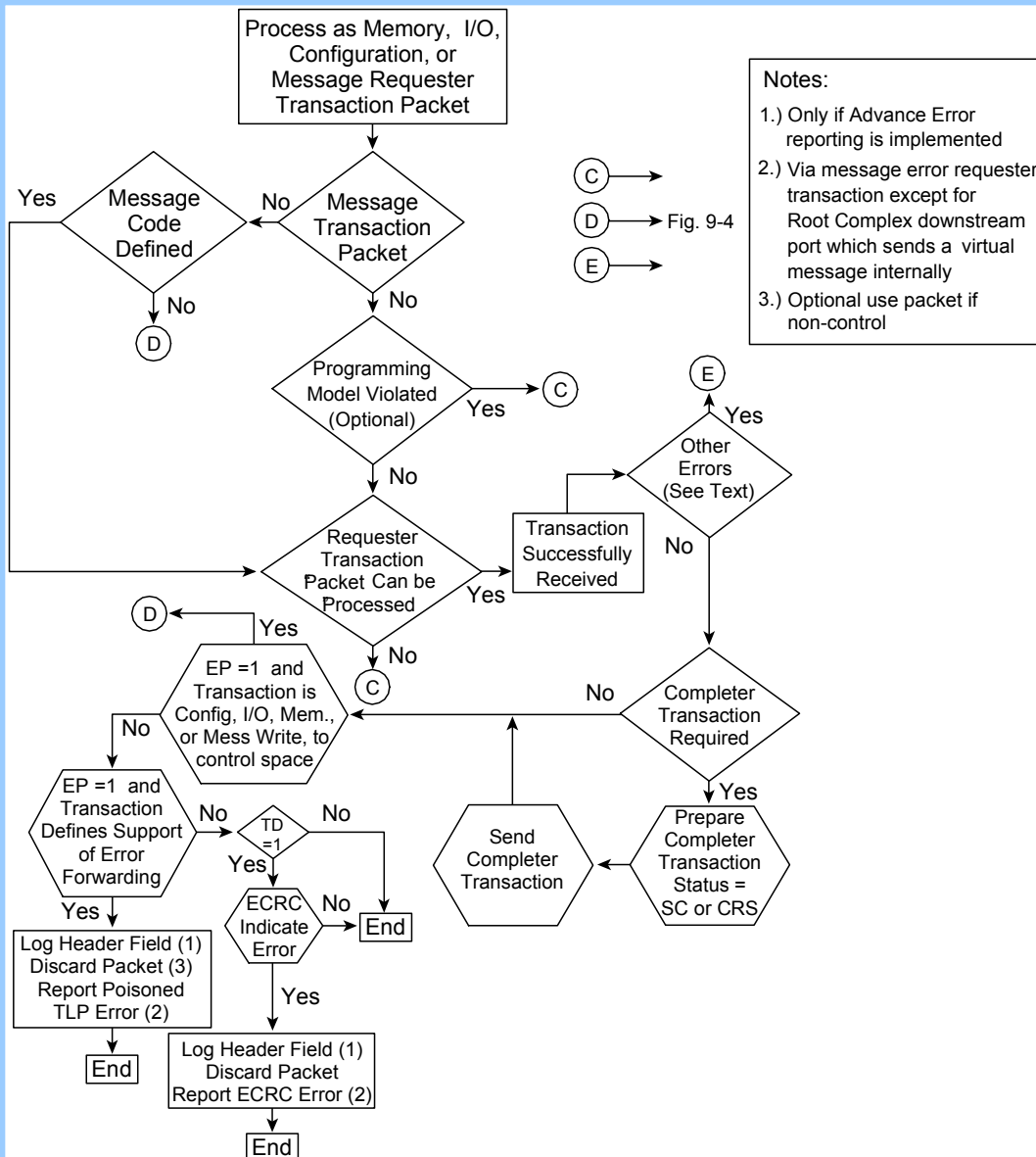
## Sample Information for Topic Group 2

- The design tools expand the layer detail **beyond** the specification to focus on the interaction between packets at each layer.
- The design tools also provide **more and specific** details than the specifications on the formats of the each type of different packets.



7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
R	FMT 1 0		4	TYPE		1	T	R	2	TC 0		R	R	R	R	
T D	E P	ATTRI 1 0		R	R	9	LENGTH									0
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0				
2 0		STATUS		B C M	11 BYTE COUNT										00	
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0				
7 TAG # 0								R	6 LOWER ADDRESS 0							

- 
- The diagram illustrates the 10GBase-KR PHY architecture, divided into a **Data Link Layer** and a **Physical Layer**.
- Data Link Layer:**
- Transmitting Path:** Data (8b 1, 8b 2, 8b 3, 8b 4) is processed by a **Retry Buffer LLTP** (receiving IDLLP), then **Add FRAMING**, and **Parallel to Serial** conversion. The output is a **Symbol Stream Relative to a Symbol Period** (indicated by a white arrow).
  - Receiving Path:** Data (8bf1, 8b 1, 8b 2, 8b 3, 8b 4, 8bf2) is processed by a **Parallel to Serial** conversion.
- Physical Layer:**
- Transmitting Port:** The **Symbol Stream** (indicated by a grey arrow) passes through **Scrambling (if Enabled)** and **8/10b Encoding**. The output is a **Symbol Stream Relative to a Symbol Period**. This is then sent to a **Transmitter** block, which is clocked by **100MHz REFCLK**. The transmitter output is a **Symbol Stream** (grey arrow) that is sent to the **Receiving Port** via **Lane# 0** and **Lane# 1**. The transmitter also receives **10b 4**, **10b 2**, and **10bf1** as inputs.
  - Receiving Port:** The **Symbol Stream** (grey arrow) is received from the **Transmitter** and sent to a **Receiver** block, which is clocked by **100MHz REFCLK**. The receiver output is a **Symbol Stream Relative to a Symbol Period** (white arrow) that is sent to the **Transmitting Port** via **Lane# 0** and **Lane# 1**. The receiver also receives **10b 2** and **10b 1** as inputs.
- Legend:**
- Grey Arrow:** Symbol Stream
  - White Arrow:** Symbol Stream Relative to a Symbol Period
- Notes:**
- 1.) Insert D and K symbols are needed for lane fillers. Also insert any SKIP Ordered Sets are needed.



# Sample Information for Topic Group 2

- The design tools provides **extensive diagrams** for processing packets, exemplified here is the only one of several flowcharts for the Transaction Layer Packet.
- A major part of processing the packets is for **error checking**. This is just one of several error checking specific diagrams unavailable in the specification.

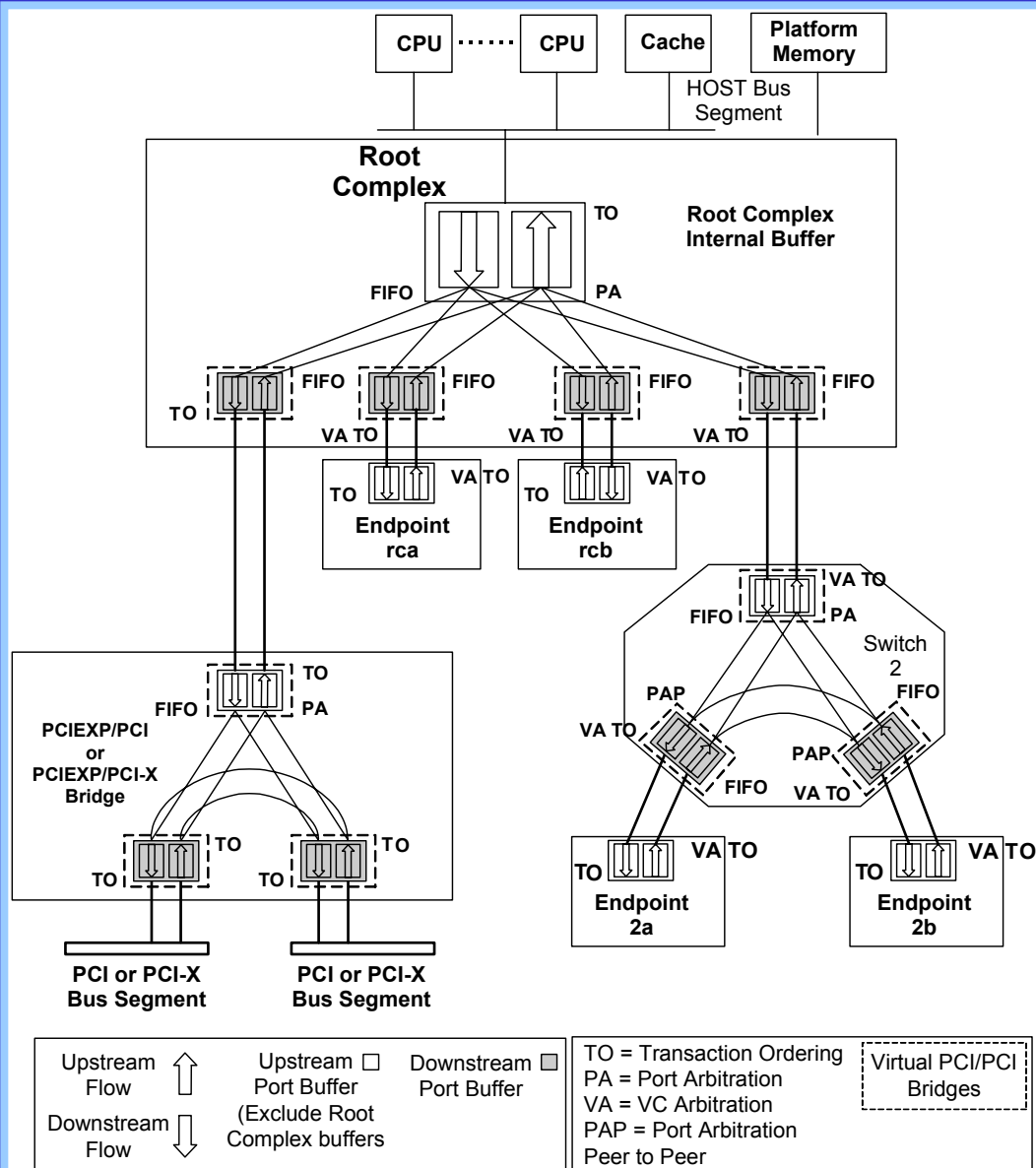
- ## Sample Information for Topic Group 2
- The design tools provides **extensive diagrams** for processing packets, exemplified here is the only one of several flowcharts for the Transaction Layer Packet.
  - A major part of processing the packets is for **error checking**. This is just one of several error checking specific diagrams unavailable in the specification.



## Sample Information from Design Tools for Topic Group 3

The requester and completers transactions between many different PCI Express devices are flowing throughout the architecture and are governed by Transaction Ordering to prevent livelock and deadlock and Flow Control Protocols provide software a means to fine tune the transaction flow in the architecture.

**Detailed Tutorial: *Transaction Ordering and Flow Control Part 1 and 2 Protocols***  
**References in the Book: *Chapters 10 to 12***

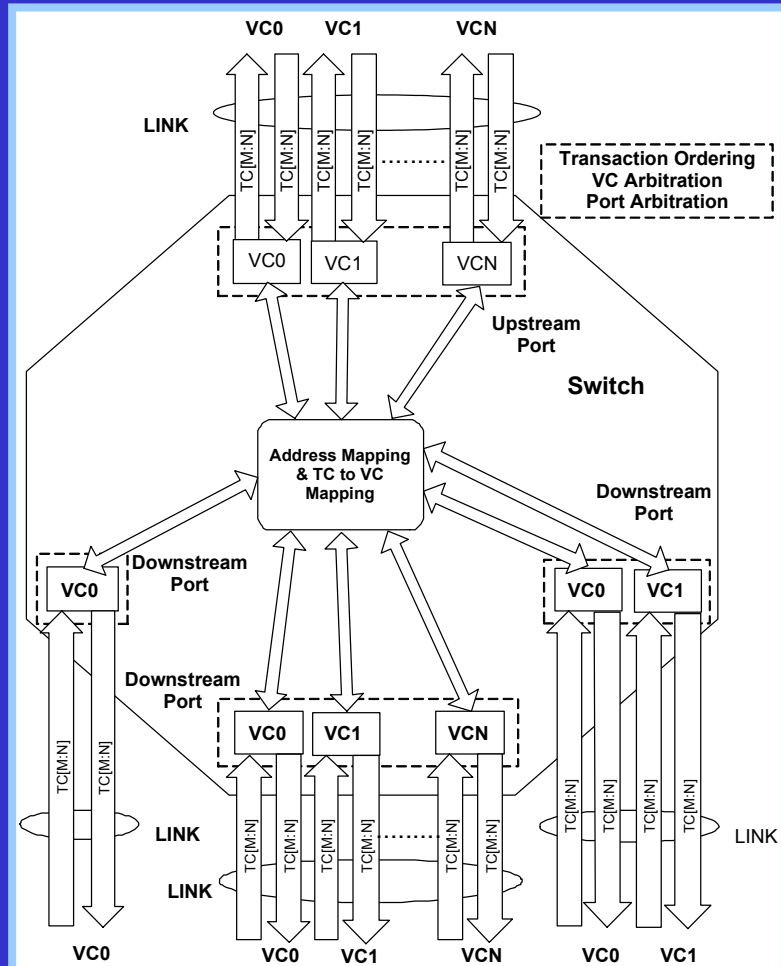
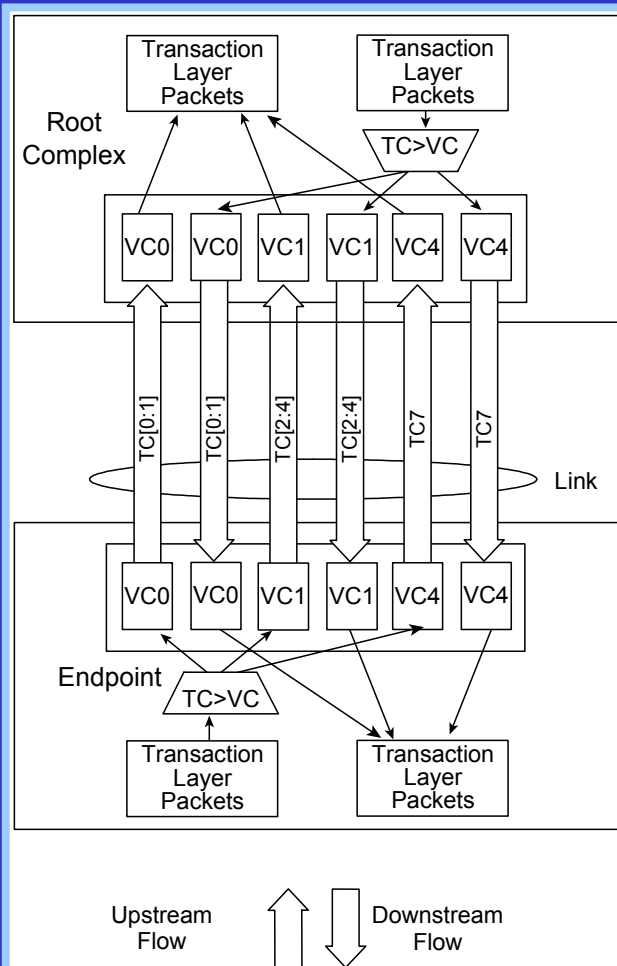


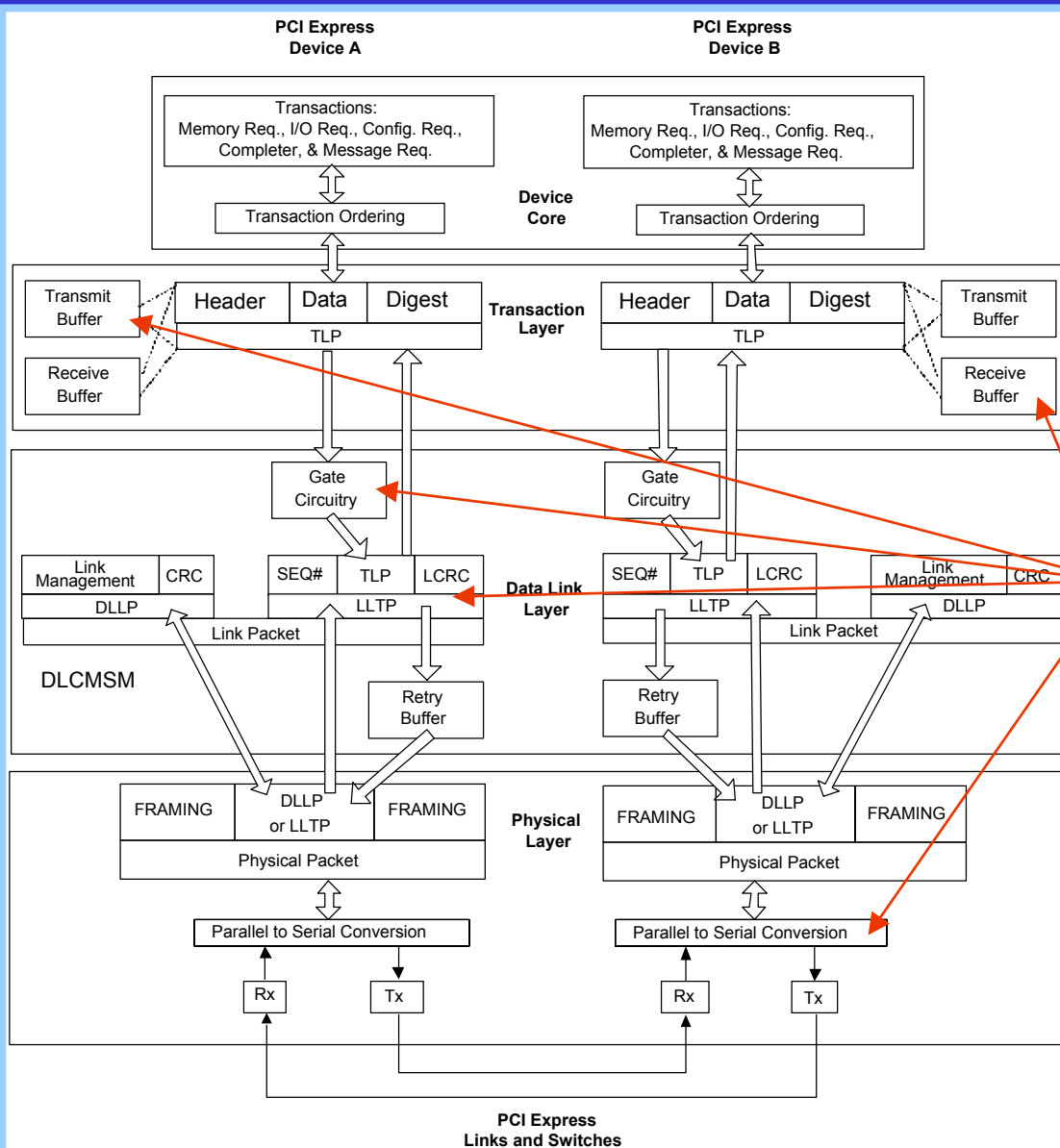
## Sample Information for Topic Group 3

- Only the design tools and not the specification **provide the full design** details for all aspects of Transaction Ordering and the Flow Control protocol relative to the buffers.
- The design tools also **clearly explain** all of the arbitration protocols for determining the next Transaction Layer Packet to transmit from each buffer.

## Sample Information for Topic Group 3

- The design tools **provide detailed** Flow Control protocol mapping, Traffic Classes, and Virtual Channels.

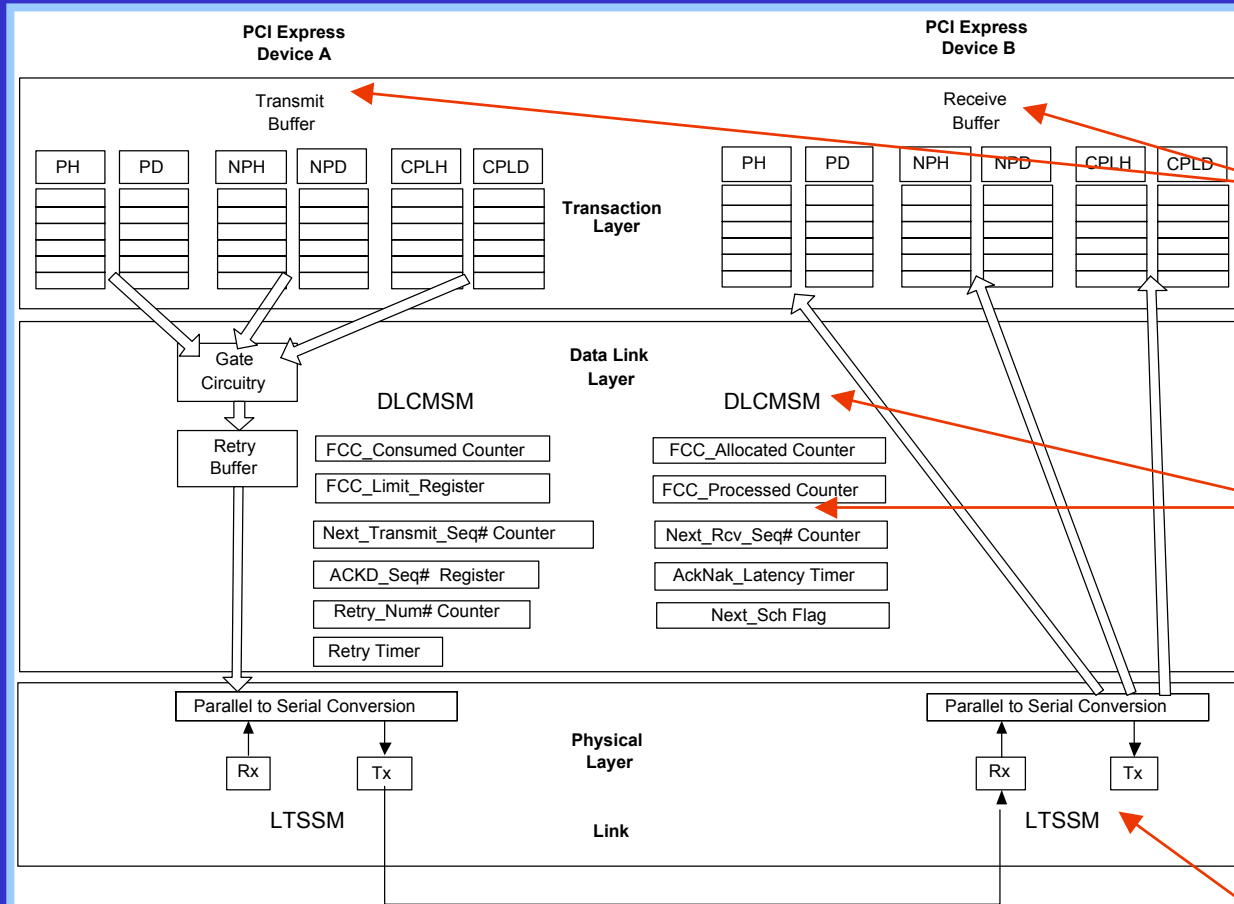




## Sample Information for Topic Group 3

- The design tools provide complete design information and extensive illustrations relative to **each hardware** element of the different layers.
- The specification do not provide any insight into the comprehensive hardware design.

## Sample Information for Topic Group 3



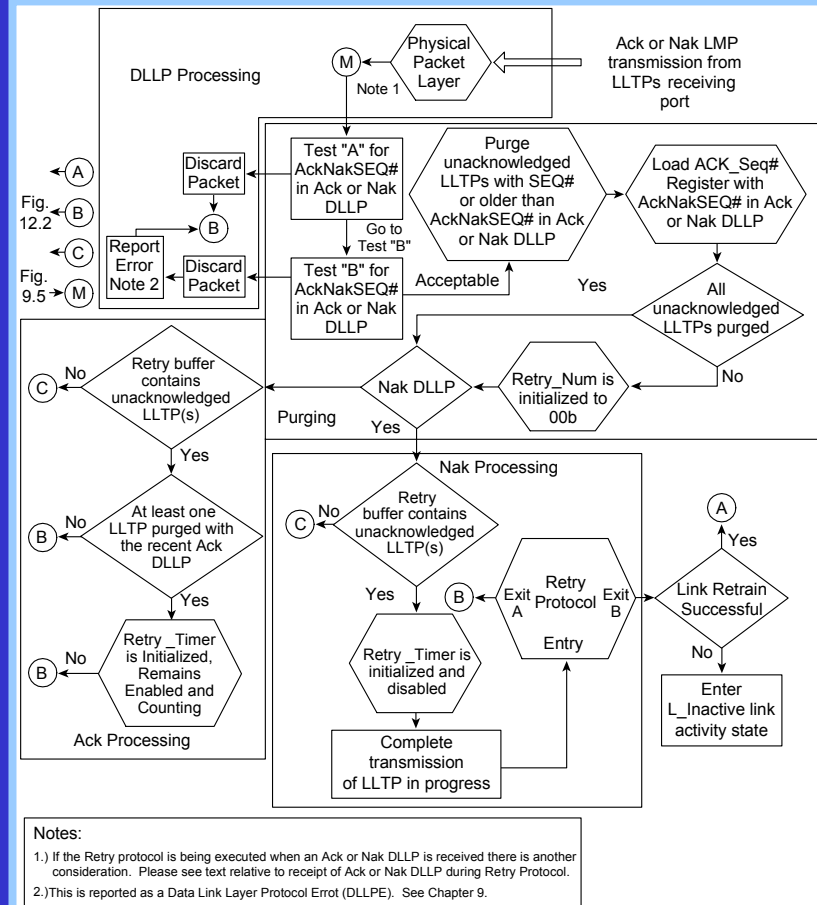
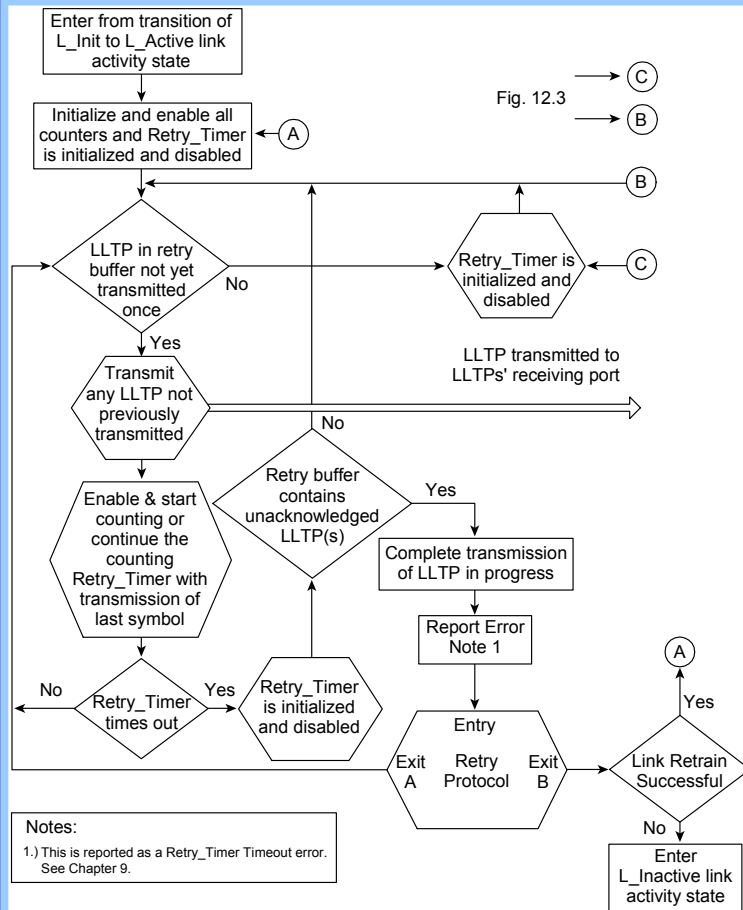
- As needed, the design tools provide **details of details**. For example, the Transmit and Receive Buffers conceptually are each a set of buffers for each specific VC number and each type of TLP.

- The logic of the Data Link Layer that controls the Gate circuitry is detailed in the **DLCMSM** (Data Layer Control and Management State Machine) in conjunction with details of all counter, timers, and flags.

- The logic of the Physical Layer is detailed **LTSSM** (Link Training and Status State Machine).

## Sample Information for Topic Group 3

- **All of the design considerations** for transmitting and receiving packets are detailed. Here are just two examples of the many flow charts in the design tools unavailable in the specifications.



## **Sample Information from Design Tools for Topic Group 4**

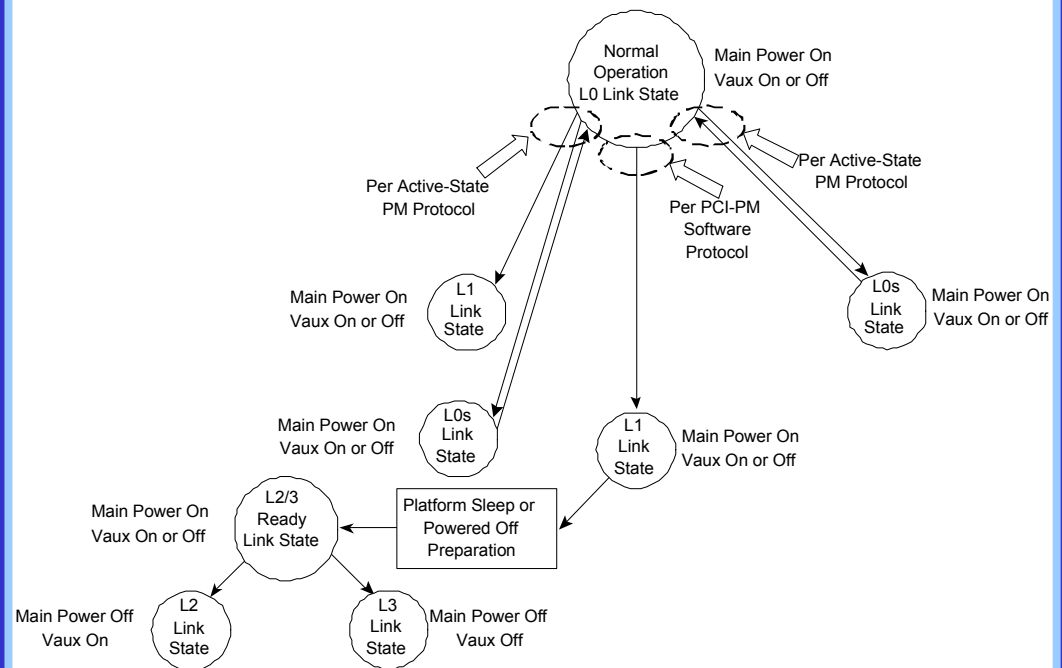
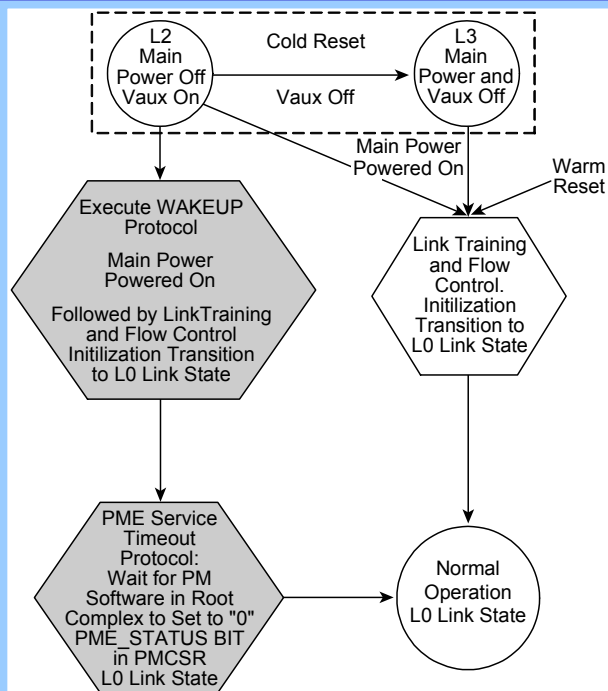
In addition to the basic operation of PCI express devices interacting with packets via the address spaces there are other interactions relative to reset and power reduction, and wake events from sleep and for Hot Plug protocol.

**Detailed Tutorial: *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

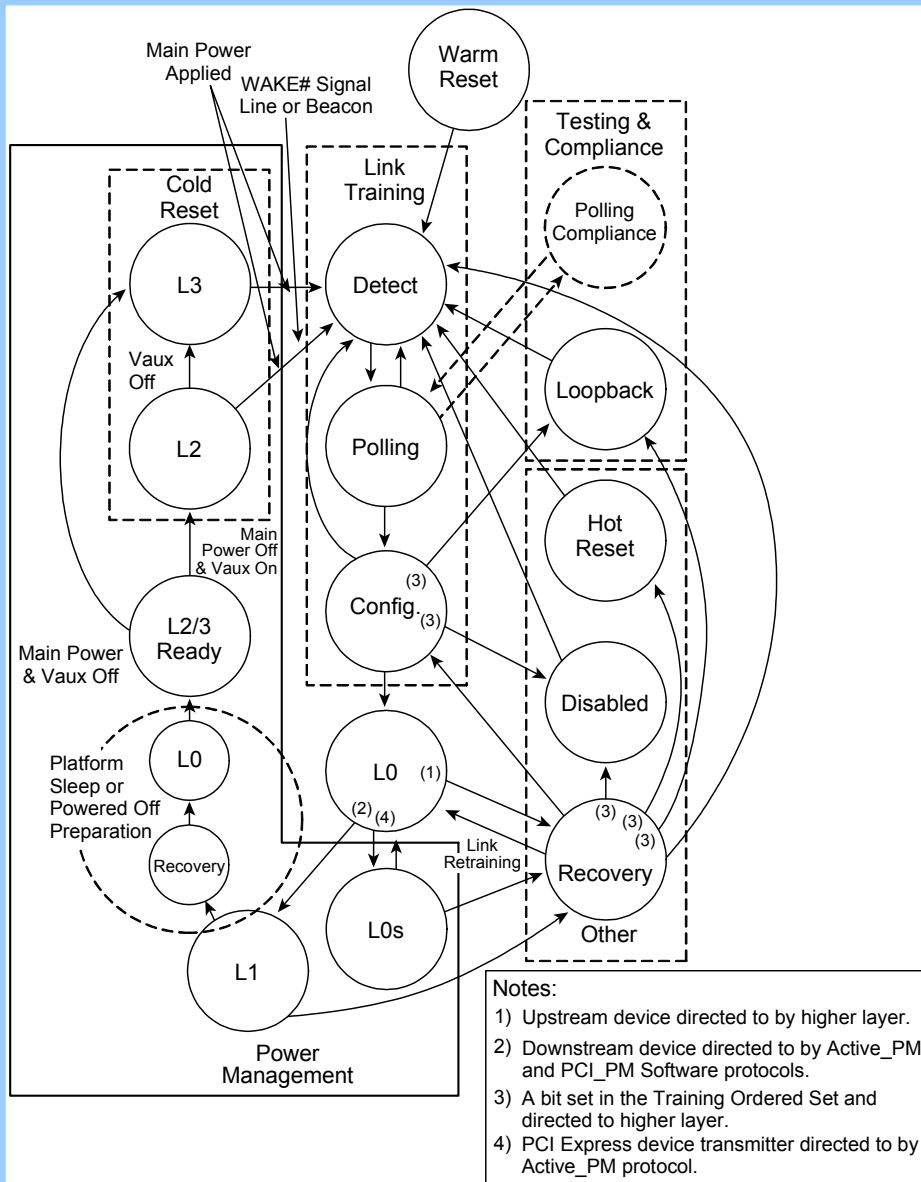
**References in the Book: *Chapters 13 to 17***

## Sample Information for Topic Group 4

- All or part of the PCI Express platform may be in sleep(L2) or powered off (L3), both are defined as Cold Reset. Also, a Warm Reset is defined. Once in the L0 link state there are two protocols to transition to a lower powered state: Active-State Power Management and PCI Power Management.
- Transition between the different power levels are **illustrated and described in details** by the design tools in details simply not available in the specification. The design tools break down the complexity of these two protocols with extensive flow charts and narrative. The examples below are only **few of the overview flow charts** from the design tools.







## Sample Information for Topic Group 4

- Each port of a PCI Express device implements the Physical Layer which contains the Link Training and Status State Machine (LTSSM).
- The LTSSM and the associated links transition between several link states.
- Only the design tools provide the **full and complete link state diagram and associated sub-link states**. The specifications only provide a general overview and does not provide all of the details.

Table 14.18: Polling.Active Link Sub-state

<b>Present State Attributes:</b> Upon entry into this link sub-state, the transmitters begin transmitting TS1 OSs with the LANE# and LINK# = PAD K symbol on <i>ALL</i> detected un-configured lanes. Physical LinkUp = "0" is indicated from Physical Layer to Data Link Layer.	
Cause of Transition to Next State	Next State Transitioned To
Within the first 24 msec. (entry timeout) upon entering this link sub-state: The transition occurs if receivers on <i>ALL</i> detected un-configured lanes successfully received eight consecutive TS1 or TS2 OS with the LANE# and LINK# = PAD K symbols (1). The transmitters in the same port as the receivers must have transmitted at least 1024 TS1 OS on <i>ALL</i> detected un-configured lanes after receiving the first TS1 or TS2 OS. The transmissions of course began with entry into this link sub-state.	Polling.Config.
After a 24 msec. upon entering this link sub-state: The transition occurs if receivers on <i>ANY</i> detected un-configured lanes successfully received eight consecutive TS1 or TS2 OSs with the LANE# and LINK# = PAD K symbols were received during entry timeout (1). The transmitters in the same port as the receivers must have transmitted at least 1024 TS1 OS on <i>ALL</i> detected un-configured lanes after receiving the first TS1 or TS2 OS during entry timeout <i>and ALL</i> detected un-configured lanes have been detected by the receivers to have exited from "electrical idle" at least once since entering this link sub-state.	Polling.Config.
After 24 msec. upon entering this link sub-state: The transition occurs if receiver on at least <i>one</i> detected un-configured lane has never detected an exit from "electrical idle" during entry timeout.	Polling. Compliance
After 24 msec. upon entering this link sub-state: The transition occurs if receivers on <i>ANY</i> detected un-configured lanes have not successfully received eight consecutive TS1 or TS2 OS (1) with the LANE# and LINK# = PAD K symbols received during entry timeout. As part of the transition the data bit rate defined by Data Bit Rate Identifier Bits [7:0] is reduced unless already at the minimum (1).	Detect.Quiet

## Sample Information for Topic Group 4

- The Physical Layer's Link Training and Status State Machine (LTSSM) contains several link states.
- All of the possible link states and link sub-states of the LTSSM and associated links are documented with exhaustive next state tables to assist in the proper design.
- **ONLY the design tools provide exhaustive Next State tables**, simply not available in the specifications.

## Sample Information from Design Tools for Topic Group 5

Part of the PCI Express protocol depends on the support of PCI compatible interrupts and lock function to retain for PCI software compatibility through the use of unique transactions and message requester transactions.

**Detailed Tutorial: *Other Hardware Topics***  
**References in the Book: *Chapters 18 to 21***

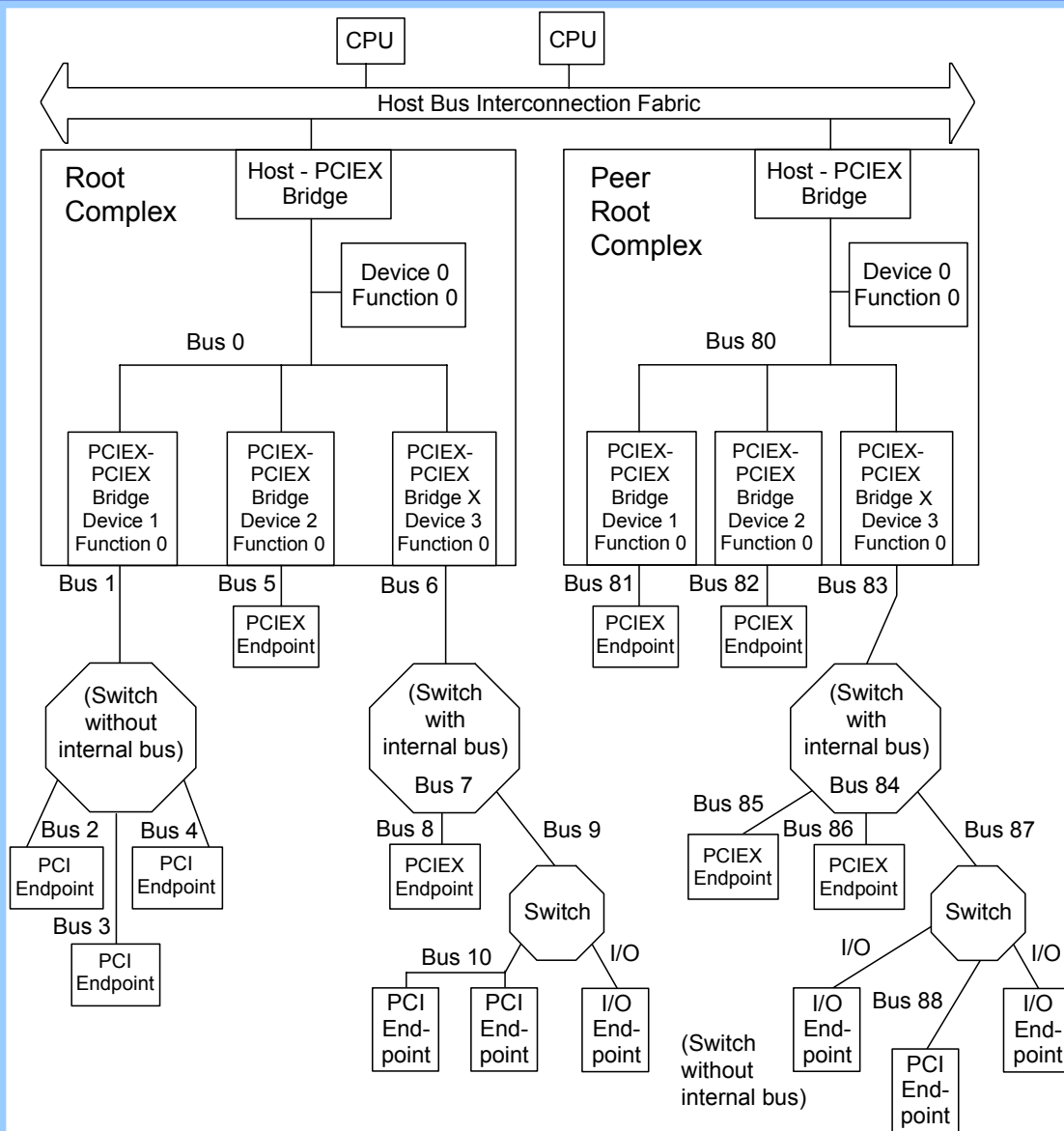
## Sample Information from Design Tools for Topic Group 5

- **Advanced Switching:** An additional inter-platform protocol can be applied on top of PCI Express protocol. The basic concept is messages can be exchanged between PCI Express devices downstream of one Root Complex with PCI Express devices downstream of another Root Complex. There may be more than two Root Complexes interconnected and the Root Complexes can be on different platforms. The advance switching protocol will be available Q1 of 2004. The **design tools provide references** to this protocol with retention of message advanced switching requester transactions and advance peer-to-peer link information. The specifications have dropped all references.
- **Interrupts:** The hardware INTx signal lines of PCI are implemented via emulation by message requester transactions. The message requester transactions emulate the assertion and deassertion of virtual interrupt signal lines. PCI Express also supports PCI compatible MSI protocol which is simply memory write requester transactions. The design tools provide **extensive step by step discussion of the emulation** not available in the specifications.
- **Lock Protocol:** The hardware LOCK# signal line of PCI is are implemented via emulation by memory requester transactions and a message requester transaction to terminate the Lock function. The inclusion of the Lock function is to support exclusive hardware access in support of PCI compatible software.
- **Mechanical and Electrical Overview:** The add-in cards sizes supported by PCI Express are essentially those supported by PCI. The mobile add-in cards are similar to PCI with enhancements to the mobile implementation. The electrical requirements focus on a differentially driven signal line pair for each lane of a link. These signal line pairs integrate a reference clock. Consequently, there is no electrical relationship with PCI or PCI-X which use CLK signal lines and strobes in parallel with signal lines.

## Sample Information from Design Tools for Topic Group 6

A key consideration of PCI Express is the compatibility to **PCI software and the registers of the configuration address space.**

Detailed tutorial: *Software Considerations*  
References in the Book: *Chapters 22 to 24*



## Sample Information for Topic Group 6

- Explanation of configuration space addressed by bus/device/function under a root complex and peer root complexes.
- Enumeration examples.
- Type 0 and Type 1 configuration accesses explained.
- Description of PCI Express upstream port configuration versus downstream port configuration and port switch requirements.
- Complete example hierarchies with PCI Express switches, bridges, and endpoints shown.

<b>Width</b>	1 byte
<b>Valid Values</b>	00h to FFh.
<b>Description</b>	This register gives the bus number on which this function resides. At reset, this value must be 00h. Configuration software writes this register when it enumerates the PCI Express hierarchy and assigns bus numbers. This register must be writeable, with one exception: The primary bus in the Root Complex is always bus zero, so functions permanently attached to that bus may hardwire this register to 00h.

<b>Width</b>	1 byte
<b>Valid Values</b>	01h to FFh.
<b>Description</b>	This register gives the bus number immediately to the south of this bridge function. At reset, this value must be 00h. Configuration software writes this register when it enumerates the PCI Express hierarchy and assigns bus numbers. This register must be writeable.  Note: This register should not be changed by system software while any devices on the secondary bus have completion transactions pending anywhere in the system.

<b>Width</b>	1 byte
<b>Valid Values</b>	01h to FFh.
<b>Description</b>	This register gives the highest bus number in the bus hierarchy south of this bridge function. At reset, this value must be 00h. Configuration software writes this register when it enumerates the PCI Express hierarchy and assigns bus numbers. This register must be writeable.

## Sample Information for Topic Group 6

- Complete register definitions describe bit-level detail, one-stop reference source. No need to leaf through multiple PCI and PCI Express specs and reference materials to gather configuration register definitions.
- Exemplified here are the following:
  - Primary Bus Number Register (Offset 18h)
  - Secondary Bus Number Register (Offset 19h)
  - Subordinate Bus Number Register (Offset 1Ah)
- Usage hints and software implications included in notes where appropriate for each register.

# The Complete PCI Express Reference Topic Group 1 Tutorial

Disclaimer: Intel, Research Tech Inc. and the authors of this tutorial make no warranty for the accuracy or use of the information No direct or indirect liability is assumed and the right to change any information without notice is retained.



## Design Tools for PCI Express

The PCI Express specification is not organized by clear design topics, does not provide sufficient details to easily master PCI Express, and does not provide in depth illustrations and flowcharts to assist in designing components. Intel recognized that simply summarizing or re-wording the specification as typically done in the other design tools in the industry is insufficient ... more extensive information is are needed.

To provide designers with PCI Express design and implementation information that is easy to master and detailed enough to assist in correct design, two design tools are available:

Six Detailed Tutorials and a new and exhaustively detailed design book.

These design tools focus on Six Quick and Easy Topic Groups which simplify the mastery of PCI Express. They save a designer weeks of trying to unravel the specification and provide the assurance of correct design implementation the first time.

## Design Tools for PCI Express

### The “Book”

The primary design tool that provides total design mastery is *The Complete PCI Express Reference* book written by Edward Solari and Brad Congdon and published by Intel ... referred to as the “Book”.

The Book provides the complete and extensive narrative of detailed figures (over 250), detailed design flow charts, and exhaustive tables for the complete understanding and design assistance in over 1000 pages. The Book can be ordered at [www.amazon.com](http://www.amazon.com) ... ISBN # 0971786194.

### Detailed Tutorials

Six free **Detailed Tutorials** ... One self paced tutorial for each of the Six Quick and Easy Topic Groups. Each introduces PCI Express information with a narrative that complements detailed figures, flow charts, and tables for each specific Topic Group from the Book. The six free **Detailed Tutorials** are available at [www.intel.com/intelpress/pciexpresscomplete](http://www.intel.com/intelpress/pciexpresscomplete).

### This Detailed Tutorial for Topic Group 1

Detailed Tutorial: *Platform Architecture and Accessing of Resources within Architecture*  
References in the Book: *Chapters 1 to 4*

## PCI Express in Six Topic Groups

### Topic Group 1

**Tutgroup1: *Platform Architecture and Accessing of Resources within Architecture***

**References in the Book: *Chapters 1 to 4***

### Topic Group 2

**Tutgroup2 : *Packets' and Layers' Specifics and Errors***

**References in the Book: *Chapters 5 to 9***

### Topic Group 3

**Tutgroup3 : *Transaction Ordering and Flow Control Part 1 and 2 Protocols***

**References in the Book: *Chapters 10 to 12***

### Topic Group 4

**Tutgroup4 : *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

**References in the Book: *Chapters 13 to 17***

### Topic Group 5

**Tutgroup5 : *Other Hardware Topics***

**References in the Book: *Chapters 18 to 21***

### Topic Group 6

**Tutgroup6 : *Software Considerations***

**References in the Book: *Chapters 22 to 24***

# Platform Architecture and Accessing of Resources within the Architecture

## Chapters 1 to 4

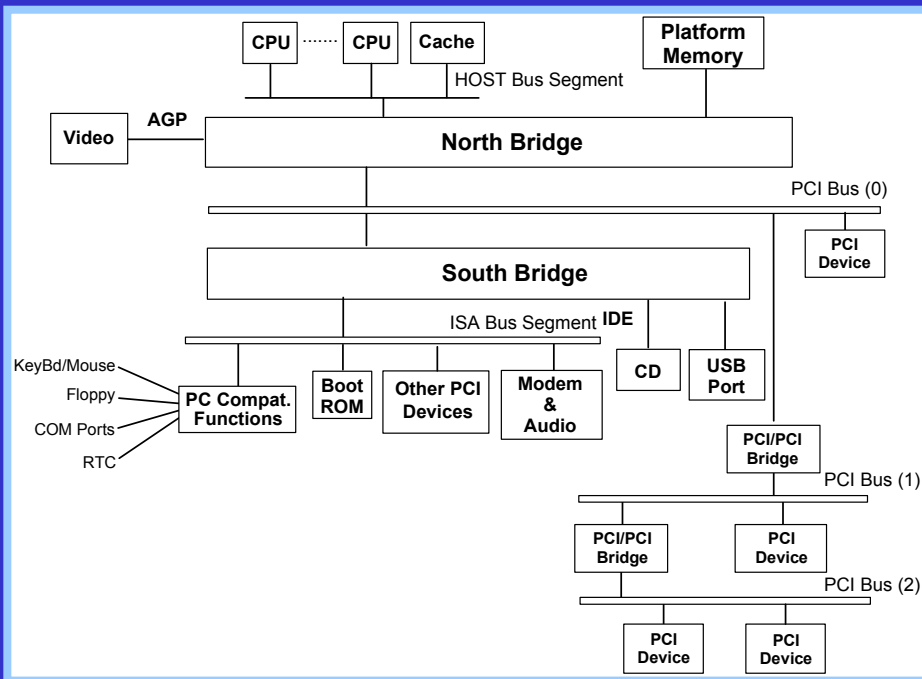
### Topic Group 1

The PCI Express **platform architecture** is an evolution of PCI and PCI-X bus segments to a **point-to-point interconnections**

**Summary:** In order to replace PCI bus segments switches are implemented with interconnecting links between PCI Express devices. PCI Express implements a **serial bit stream** to minimize wire count of the interconnecting links and transfer **packets** between PCI Express devices. The packets contain transactions relative to the **memory, I/O, and configuration address spaces** retained from PCI and the message address space. The **message address space** is unique to PCI Express and replaces unique PCI bus transactions and hardware signal lines. The interaction between PCI Express devices is done in two parts per the **Requester/Completer protocol** implementing requester and completer transactions

# Chapter 1

## Architecture Overview



## PCI Platform Architecture and Performance

- PCI platforms evolved into North and South bridges represented by Memory and I/O Controller Hubs
- Primary concept is that high performance resources are connected to the Memory Controller Hub and the lower performance resources are connected to the I/O Controller Hub.
- Within the lower performance resources of the I/O Controller Hub, some require a high performance connection (e.g. CD) and others can use a shared PCI bus segment (e.g. Ethernet and SCSI)

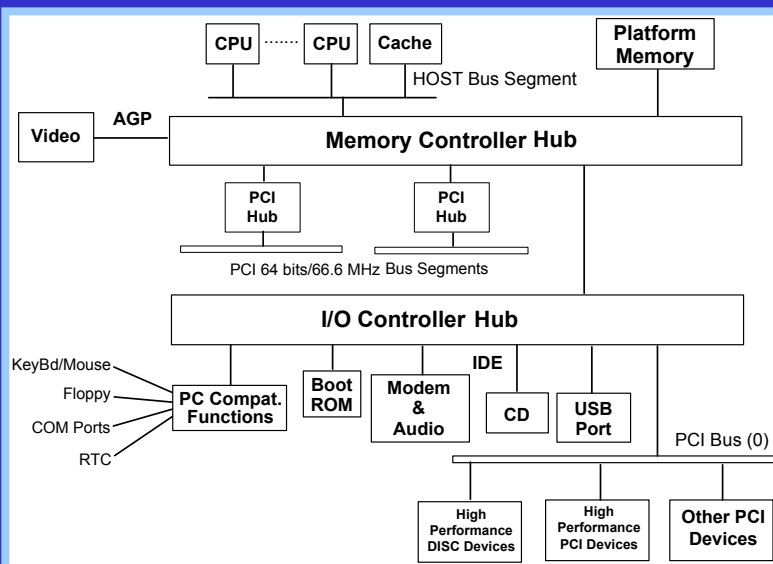


Table 1.1: PCI Bus Bandwidth and Add-in Card Slot Limitation

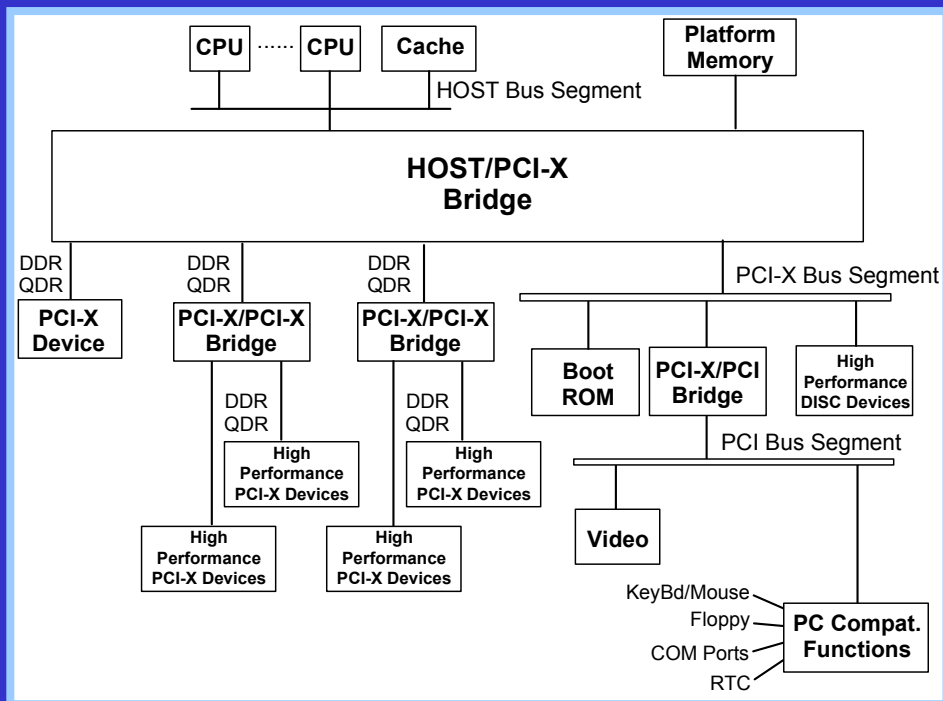
Data Bus Width	Signal Lines (excluding power, arbitration, and test)	Clock Signal Line Frequency (MHz)	Maximum Bandwidth (Megabytes/second)	Maximum Add-in Card Slots
PCI 32 bits	49	33.3	133.2	4 (1)
PCI 32 bits	49	66.6	266.4	2 (1)
PCI 64 bits	81	33.3	266.4	2 (1)
PCI 64 bits	81	66.6	532.8	2 (1)

Table Notes:

- 1) Other platform components in addition to the bridge or add-in slot are possible, but the entry is the limit of add-in cards with one platform bridge and one other platform component.

## PCI Platform Architecture and Performance ... continued

- As technology evolved there was a greater need to support more high performance resources. The I/O Controller Hub did not provide sufficient performance for all high performance resources relative to Platform Memory. Additional high performance PCI segments were added to the Memory Controller Hub.
- In addition to the connection to the Memory Controller Hub, the PCI bus segment size and frequency was increased.
- At 64 data bits PCI achieves bandwidths of 532.8 Megabytes/second.
- It is also possible to integrate the two Hubs into a single Host/PCI Bridge. The balance of this tutorial will assume an a single HOST/PCI or HOST/PCI-X Bridge.



## PCI-X Platform Architecture and Performance .. continued

- PCI-X was developed to extend performance beyond PCI.
- The HOST/PCI-X Bridge in this discussion represents a consolidation of the Hub controllers into a single bridge structure.
- PCI -X initially simply increased the CLK signal line frequency over PCI to increase bus segment bandwidth. Eventually, PCI-X DDR and QDR provided source synchronous strobes to improve bus segment bandwidth. “D” and “Q” refers to the two and four strobe points within a single CLK signal line period, respectively. Under future consideration is PCI-X 3.0 with source synchronous.
- The increase in bus segment bandwidth greatly reduces the number of add-in card slot per bus segment.
- The resulting higher performance PCI-X DDR and QDR results in point-to-point interconnections.



Table 1.3: PCI-X Bus Bandwidth and Add-in Card Slot Limitation

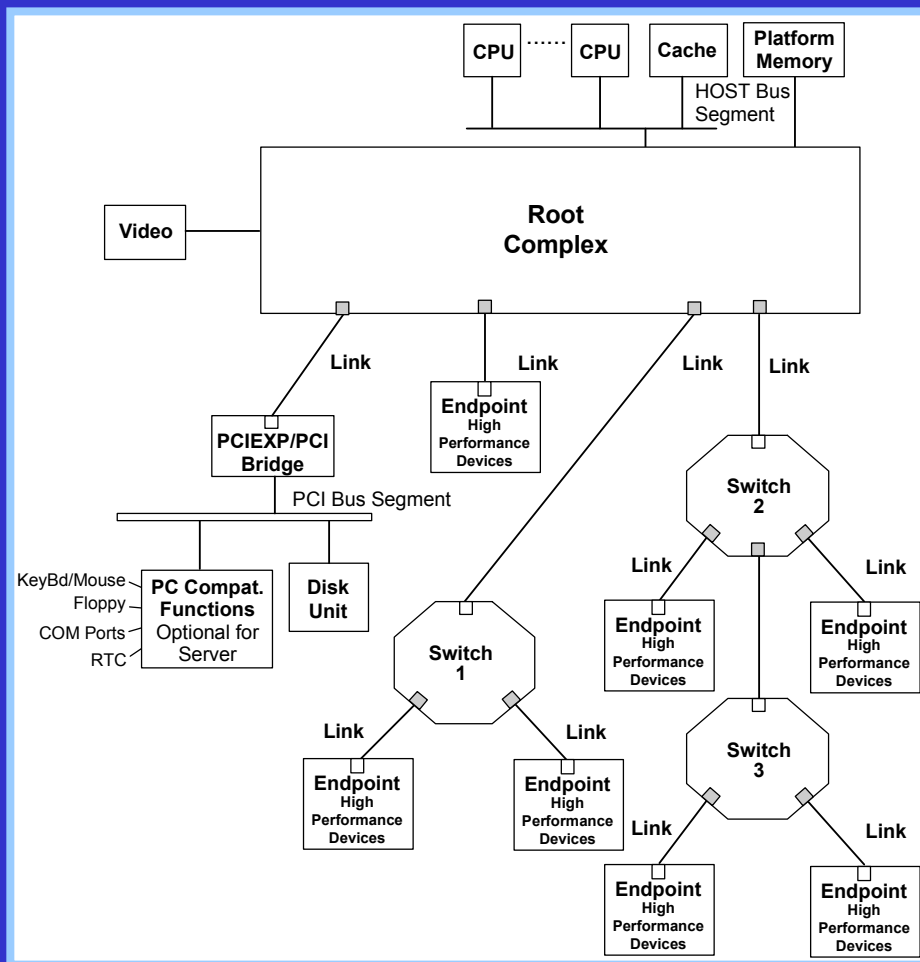
Data Bus Width	Signal Lines (excluding power, arbitration, and test)	Clock Signal Line Frequency (MHz)	Maximum Bandwidth Megabytes/Second	Maximum Add-in Card Slots
PCI-X 1.0 64 bits	82 lines	66.6	532.8	4 (1)
PCI-X 1.0 64 bits	82 lines	100	800	2 (1)
PCI-X 1.0 64 bits	82 lines	133.2	1065.6	1 (1)
PCI-X 2.0 64 bits	82 lines	Source synchronous DDR (3)	2131.2	1 (2)
PCI-X 2.0 64 bits	82 lines	Source synchronous QDR (3)	4262.4	1 (2)
PCI-X 3.0 64 bits	82 lines (4)	Source synchronous (3)	8524.8	1 (2)

Table Notes:

- 1) Other platform components in addition to the bridge or add-in slot are possible, but the entry is the limit of add-in cards with one platform bridge and one other platform component.
- 2) Essentially, the add-in card is the only item connected to the platform bridge of this associated bus segment.
- 3) PCI-X 266 and PCI-X 533 implements source synchronous for data via dual use of C/BE# signal lines. Implementation per function for PCI-X 1066 is pending.

## PCI-X Platform Architecture and Performance ... continued

- At 64 data bits PCI-X achieves bandwidths over 8524.8 Megabytes/second. This a 16 times improvement over PCI.
- The large signal line count defined for a bus segment shared among several PCI-X devices is impractical for point-to-point interconnections.



## PCI Express Platform Architecture

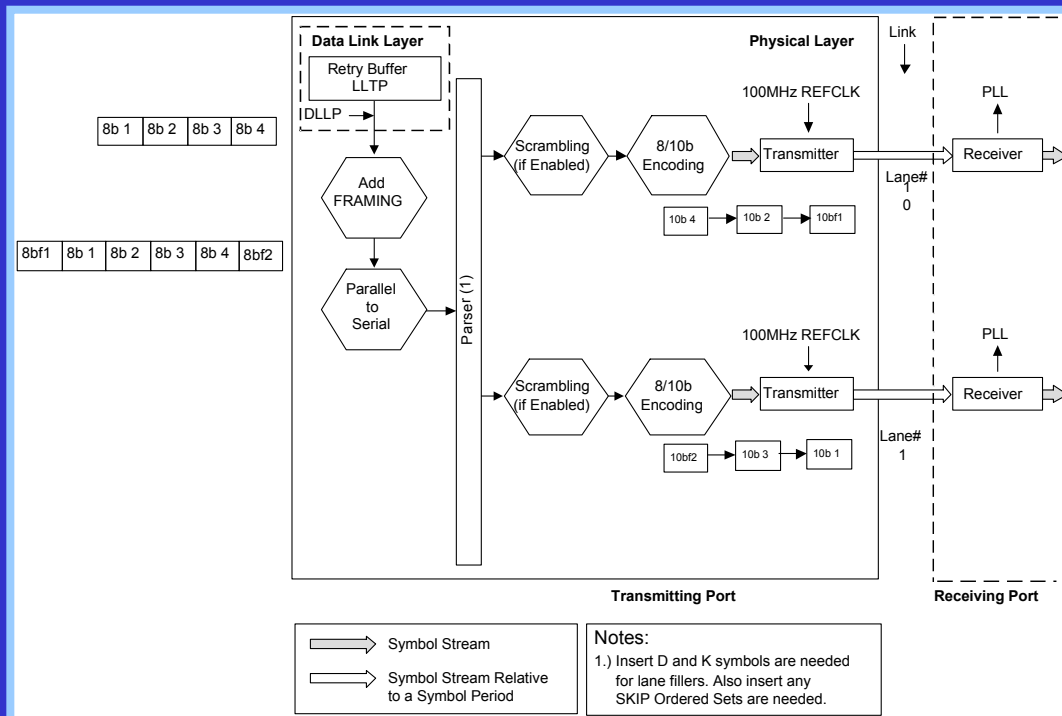
- PCI Express recognized that PCI had limited performance and PCI-X improved performance resulted in impractical point-to-point interconnections with too many signal lines.
- PCI Express recognized that high performance PCI like platforms require a point-to-point interconnection. Point-to-point interconnections based on the large signal line count of PCI or PCI - X bus segments are impractical.
- The basic elements of the PCI and PCI-X platforms are replaced by PCI Express devices and links collectively called the PCI Express fabric.

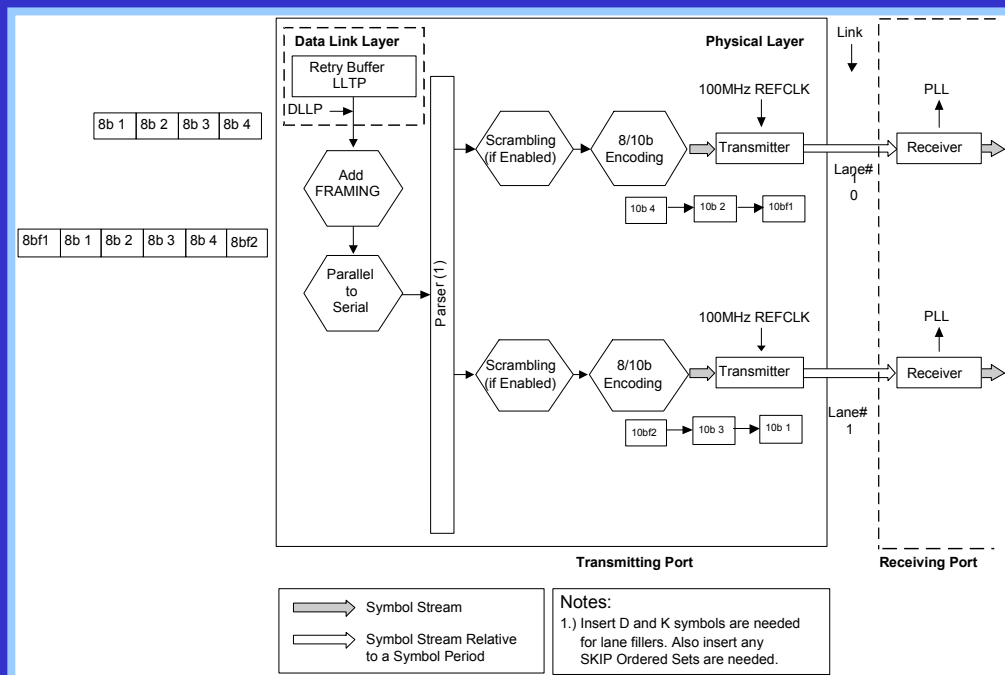
### PCI Express Fabric Nomenclature

- **Root Complex:** Replacement for Hub Controllers, Host PCI/Bridges, and Host/PCI-X Bridges.
- **Switches:** Replacement for PCI and PCI-X like bridges to provide a point-to-point interconnections between the Root Complex, endpoints, and bridges.
- **Endpoints:** PCI Express replacement for PCI and PCI-X bus masters and targets.
- **Bridges:** Interface between a PCI Express links and hierarchies of PCI or PCI-X bus segments. Typically PCIEXP (PCI Express)/PCI bridge implements PCI bus segments.

## PCI Express Fabric Nomenclature ... continued

- **Links and Lanes:** PCI Express implements point-to-point interconnections called *links*.
  - Each link consists of one or more lanes and each *lane* contains two pairs of differentially driven signal lines.
  - Each pair of differentially driven signal lines requires the packets between PCI Express devices to be transmitted across the link in a serial bit stream. The serial bit is grouped into 10 bit symbols to create a stream of symbols.
  - Per each lane, one pair of differentially driven signal lines is for packets flowing upstream and the other pair is for packets flowing downstream simultaneously.
  - When a link contains multiple lanes, the packets are parsed across the multiple lanes to increase link bandwidth.





## PCI Express Fabric Nomenclature ... continued

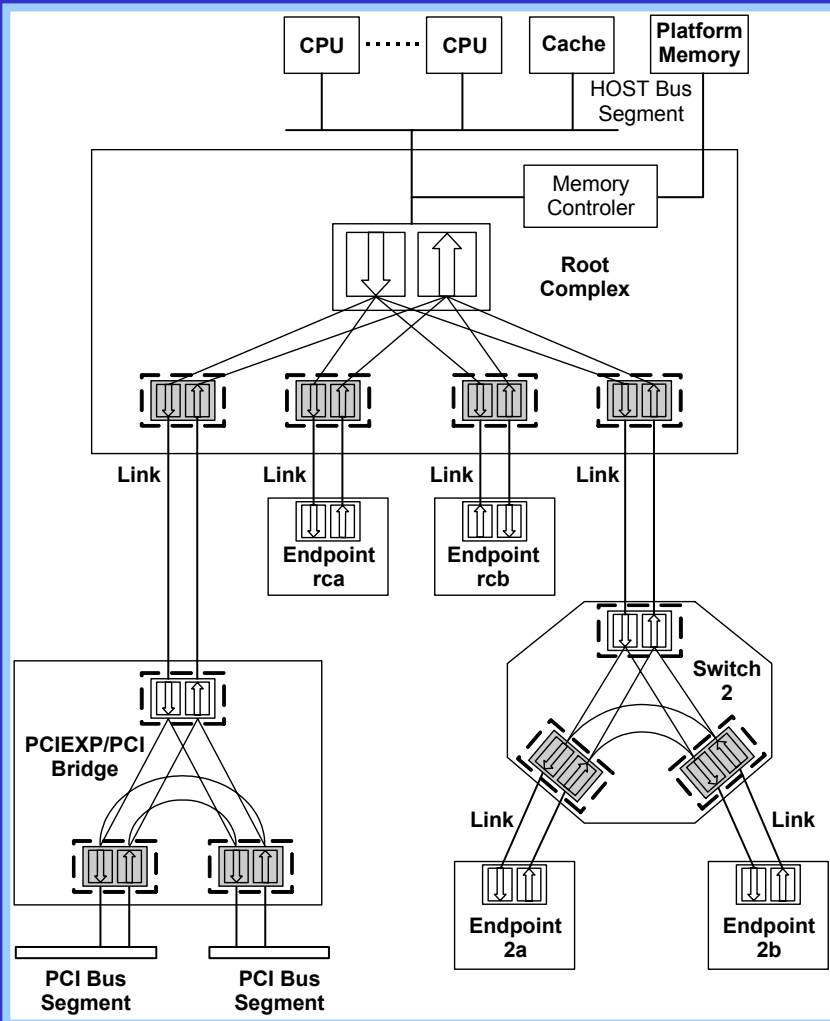
- Each link supports 1 to 32 lanes. Only two lanes are shown in this example.
  - A link between any two PCI Express devices consist of 1, 2, 4, 8, 12, 16, or 32 lanes (x1, x2, ...x32).
  - Part of what is called Link Training is to determined how many lanes can be configured in the link.
  - The number of signal lines contained in 1 to 32 lanes is 4 to 128 simultaneously supporting upstream and downstream flowing packets.
  - The number of lanes are further restricted if the application is a mobile add-in card. The associated connector does not support all possible number of lanes for a link.

PCI Express Bit Stream Rate versus Bandwidth Default 2.5Gb/sec per Differential Pair (Each Direction)							
Lanes Per Link	x1	x2	x4	x8	x12	x16	x32
Signal Lines Each direction	2	4	8	16	24	32	64
Raw Bit Stream per second	2.5 Gb/s	5.0 Gb/s	10.0 Gb/s	20.0 Gb/s	30.0 Gb/s	40.0 Gb/s	80.0 Gb/s
Bandwidth Bytes per second	250 MB/s	500 MB/s	1000 MB/s	2000 MB/s	3000 MB/s	4000 MB/s	8000 MB/s

## PCI Express Platform Performance

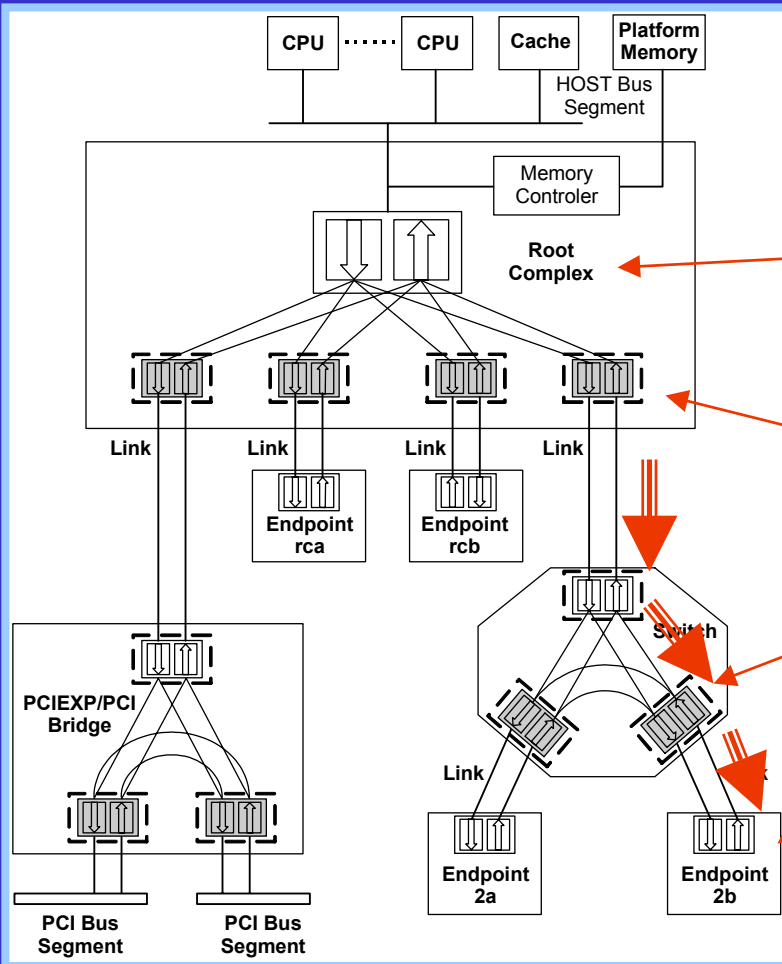
- The maximum PCI bandwidth of 532.8 Megabytes/second is approximately the bandwidth of a x2 PCI Express link. The maximum bandwidth of PCI Express is over 15 times faster than maximum bandwidth of PC.
- The maximum PCI-X bandwidth of 8524.8 Megabytes/second is approximately the bandwidth of a x32 PCI Express link. The maximum bandwidth of PCI Express is over 1.87 times faster than maximum bandwidth of PCI-X QDR. PCI Express bandwidth is about the same as the proposed PCI-X 3.0 with source synchronous.
- **However ...** The PCI Express bandwidth numbers discussed above reflect the transfer rate in each direction. The full simultaneous bi-directional bandwidth on a link is twice the numbers listed. PCI and PCI-X bus segments do not support simultaneous bi-directional bandwidth.
- The PCI Express bandwidth numbers discussed above reflect a data bit rate of 2.5 Gigabits /second. The data bit rate is the transfer rate across a link for the stream of bits on each lane that comprise the symbol stream. Future PCI Express data bit rates will be a greater.





## Requester /Completer (R/C) Protocol

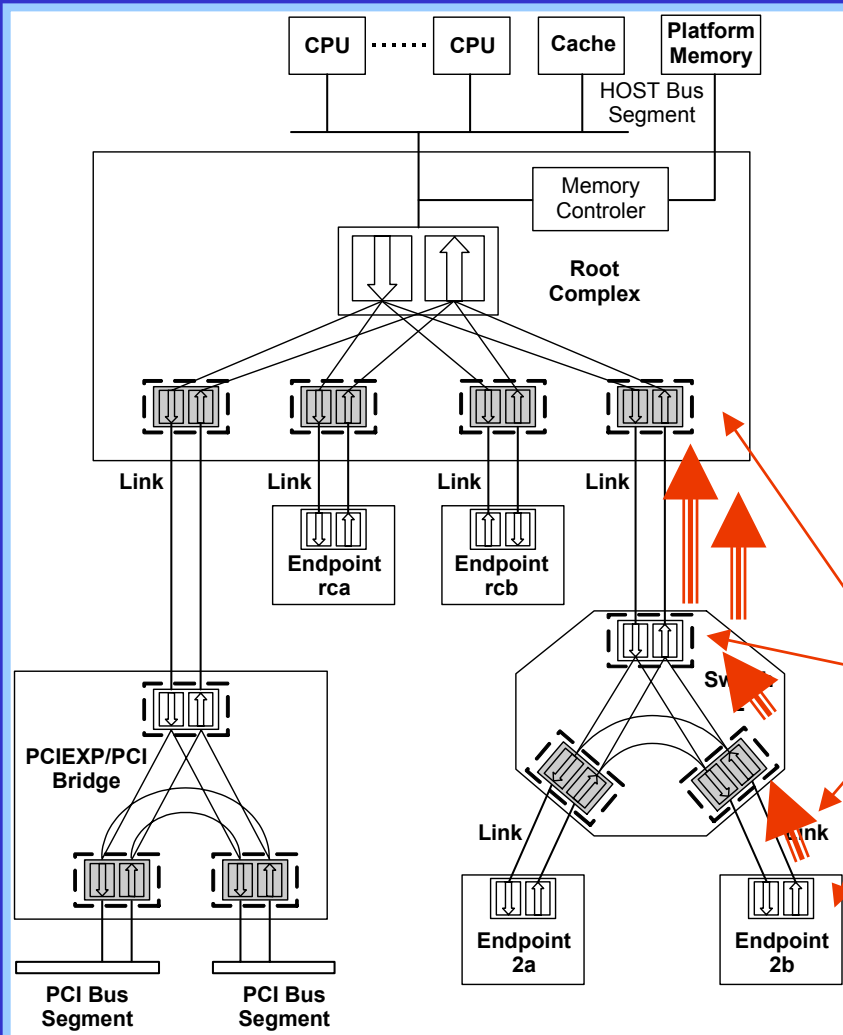
- R/C protocol is applied to all PCI Express transactions
- All PCI Express transactions consist of two parts: requester transactions and completer transactions.
  - Requester transactions provide address or routing information, and data when applicable.
  - Completer transactions provide the completion status and data if applicable
- PCI Express transactions that consists of requester transactions and associated completer transactions for the following:
  - Memory read
  - I/O Read and Writes
  - Configuration Read & Write
- Other PCI Express transactions consist of only requester transactions:
  - Memory write
  - Messages
- With the PCI Express platform there are multiple buffers associated with the links that interconnect the PCI Express devices. The requester and completer transactions move from a buffer at one of a link when sufficient space is available at the other end.
- Consequently, a transaction go from buffer to buffer until it arrives at its destination.



## Requester /Completer (R/C) Protocol

- Protocol Sequence ... Expanded
  - The PCI Express device core begins the execution of a transaction in the PCI Express device. The PCI Express device is called the **requester source**. In this discussion assume the the CPU is reading data from endpoint 2b. The Root Complex presents the CPU.
  - The **requester source** transmits the requester transaction in a Transaction Layer Packet (TLP) into the PCI Express fabric. The requester transaction is buffered until it is transmitted.
  - Per R/C protocol the TLP from the **requester source** are posted in each switch as it flows downstream.
  - The final destination of the TLP is a PCI Express device called **requester destination**. At the **requester destination** the TLP can be posted.
  - At the **requester destination** the completer transaction is composed when PCI Express device core is ready to respond.
  - If the TLP contains a memory write or message transaction, there is no completer transaction to be composed.



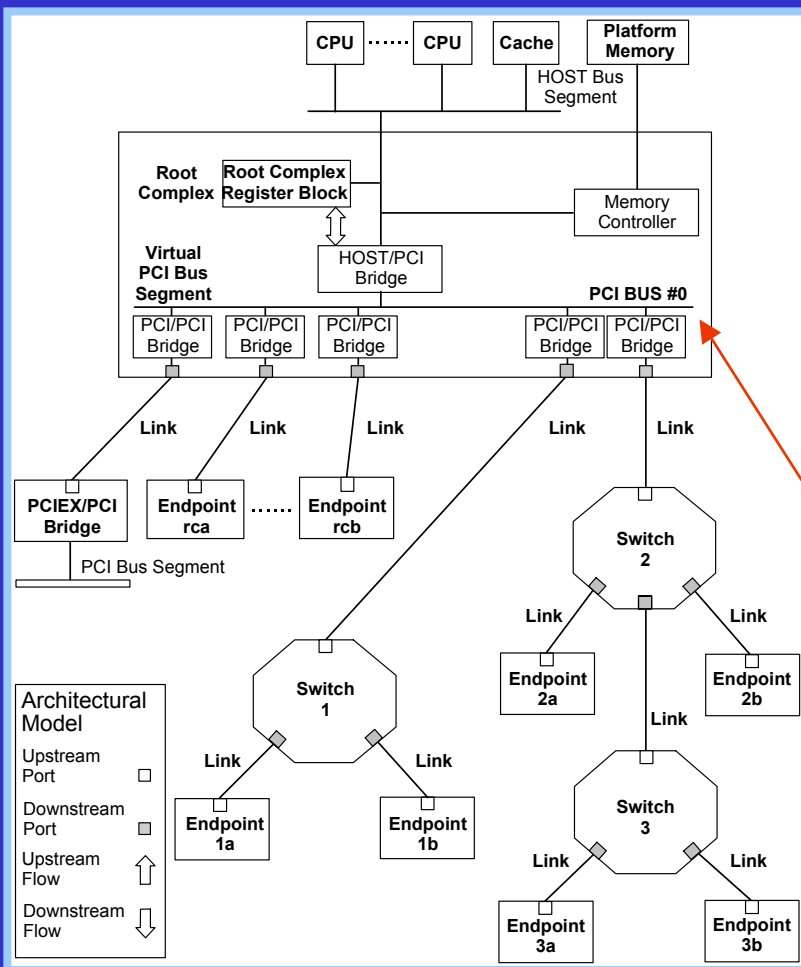


## Requester /Completer (R/C) Protocol

... continued

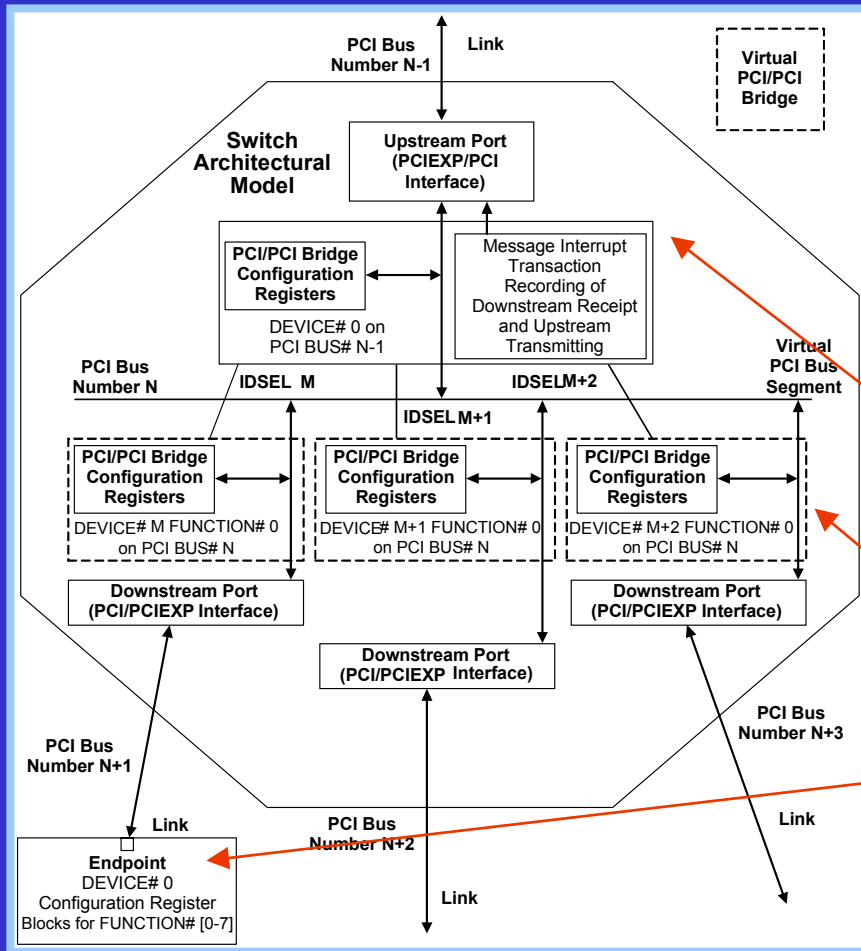
- Protocol Sequence ... Expanded
  - The **requester destination** then becomes the source of the completer transaction and is called the **completer source**. The completer transaction is buffered until it is transmitted.
  - The **completer source** transmits the completer transaction in a TLP into the PCI Express fabric
  - Per R/C protocol the TLP from the **completer source** is posted in each switch as it flows upstream
  - The final destination of the TLP is PCI Express device called **completer destination** which is also the **requester source**.

Chapters 2 & 3  
PCI Express  
Architecture Overview and Implementation



## PCI Express Platform Introduction

- The PCI devices are the Root Complex, switches, endpoints, and bridges.
- Internally all of the PCI Express devices are comprised of PCI compatible elements to retain compatibility with PCI configuration address space and software.
- Externally all PCI Express devices are interconnected by links within the PCI Express fabric.
- The link interconnections are similar to PCI bus segments in the **downstream and upstream flow of packets**:
  - The Root Complex implements virtual PCI/PCI Bridges in downstream ports. Each link is assigned a **BUS#** like PCI bus segments.



## PCI Express Platform Introduction ... continued

- The link interconnections are similar to PCI bus segments in the **downstream and upstream flow of packets ... continued**:
  - Each switch's and bridge's upstream port is connected internally to a PCI /PCI Bridge. Each of the switch's or bridge's upstream port appears as a PCI DEVICE# 0 on the link.
  - The switches implement virtual PCI/PCI Bridges in downstream ports. Each link is assigned a BUS# like PCI bus segments.
  - Each endpoint's upstream port appears as a PCI DEVICE# 0 on the link.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
R	FMT 1 0		4		TYPE		1	L	R	2 TC 0		R	R	R	R		
T D	E P	ATTRI 1 0		R	R	9		LENGTH								0	
7 BUS NUMBER								0	4 DEVICE # 0				2 FUNC# 0				
7 TAG								0	3 LAST BE 0				3	FIRST BE 0			
31 ADDRESS 16																	
15 ADDRESS 02 R R																	

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4		TYPE		1	L	R	2 TC 0		R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
STATUS 2 0		B C M		11 BYTE COUNT 00											
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG # 0								R	6 LOWER ADDRESS 0						

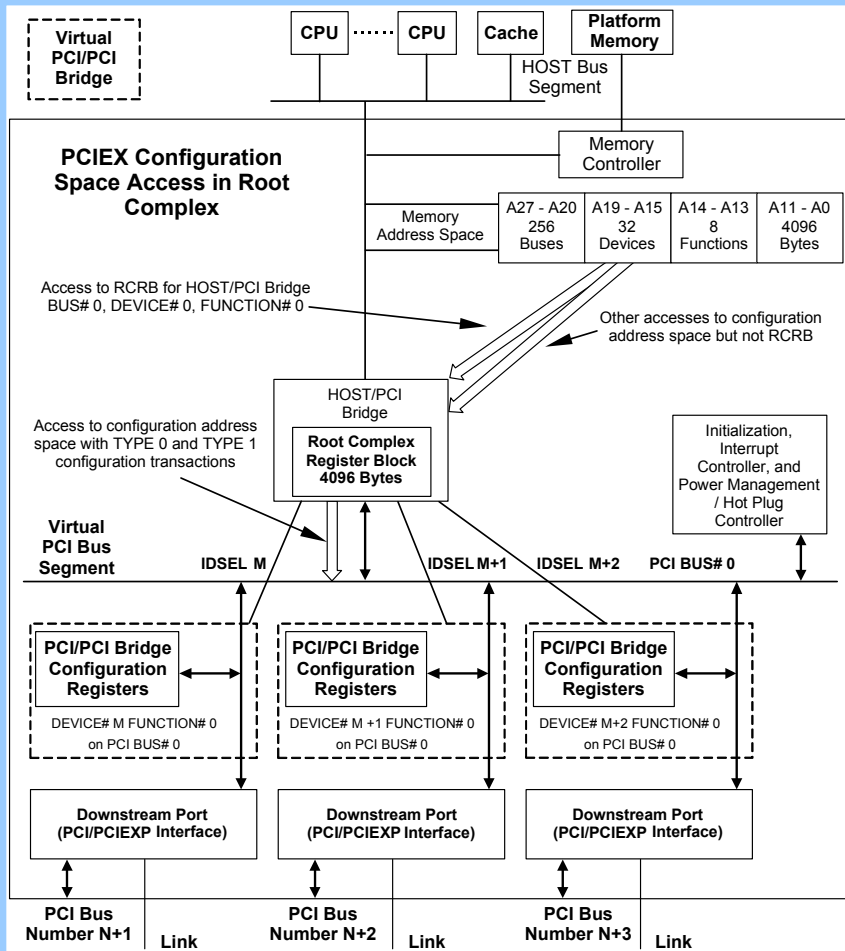
## PCI Express Platform Introduction ... continued

- The link interconnections are similar to PCI bus segments in the **downstream and upstream flow of packets ... continued:**
  - The flow of requester transactions is directed to a specific downstream port by the address in the Transaction Layer Packet (TLP) as interpreted by the switches along the path ... exemplified by the memory read requester transaction packet
  - The flow of completer transactions is directed to a specific downstream port by the ID Routing (BUS#, DEVICE#, and FUNCTION#) in the Transaction Layer Packet (TLP) as interpreted by the switches along the path ... exemplified by the I/O write completer transaction packet.
  - This ID information simply identifies the source of the packet.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
R	FMT 1 0		1	0	2rFIELD 0			R	2	TC	0	R	R	R	R				
T D	E P	ATTRI 1 0		R	R	9LENGTH0													
7BUS NUMBER0								4	DEVICE # 0				2 FUNC# 0						
7TAG0								7MESSAGE CODE0											
63								ADDRESS HDW								48			
47								ADDRESS HDW								32			
31								ADDRESS HDW								16			
15								ADDRESS HDW								00			

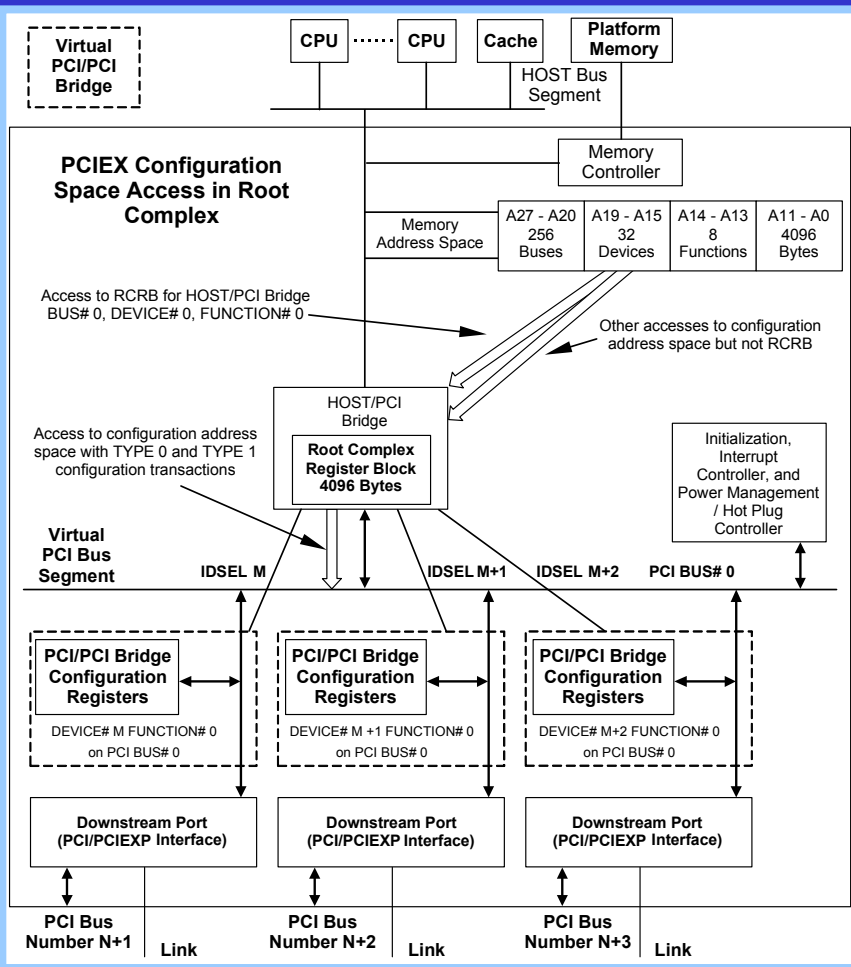
### PCI Express Platform Introduction ... continued

- The link interconnections are similar to PCI bus segments in the **downstream and upstream flow of packets ... continued:**
  - TLPs containing message transactions use Implied Routing instead of address or ID Routing. Implied Routing defined predetermined destinations as interpreted by the switches along the path.. Under to be defined implementations the address may also be used.
  - This ID information simply identifies the source of the packet.
- Peer-to peer transactions** implement addresses in the TLP to flow upstream and the downstream as exemplified in TLP from endpoint 3b to endpoint 3a.



## Root Complex Detail

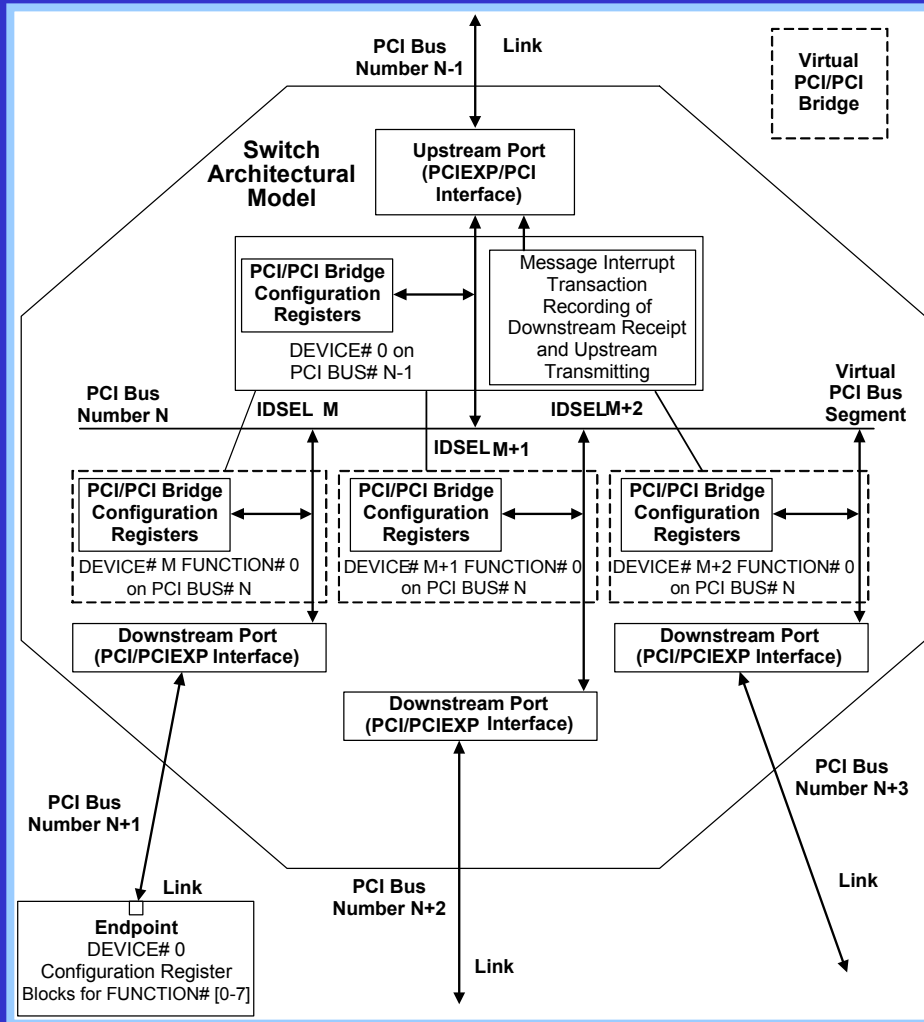
- The typical implementation of the Root Complex consists of virtual HOST/PCI and PCI/PCI Bridges, and virtual PCI bus segments.
- The virtual HOST/PCI Bridge converts Host bus segment transaction to PCI “like” bus transaction. The HOST /PCI Bridge also converts PCI “like” transactions to access Platform Memory. The Root Complex independently connects the HOST bus segment to the Platform Memory.
- Each downstream port on the Root Complex is an independent virtual PCI/PCI Bridge. Each downstream port executes the following:
  - Converts PCI “like” transactions flowing downstream into PCI Express transactions. If the PCI “like” transactions are configuration TYPE 0, the virtual PCI/PCI Bridges direct the transactions to their configuration register blocks.
  - Converts PCI Express transactions flowing upstream into PCI “like” transactions to provide access to the Platform Memory.
- The Root Complex Register Block (RCRB) is configuration registers for the virtual HOST/PCI Bridge
- For a Root Complex executing PCI compatible software the size of the RCRB is 256 Bytes and the access is via the I/O address space of the HOST bus segment.



## Root Complex Detail ... continued

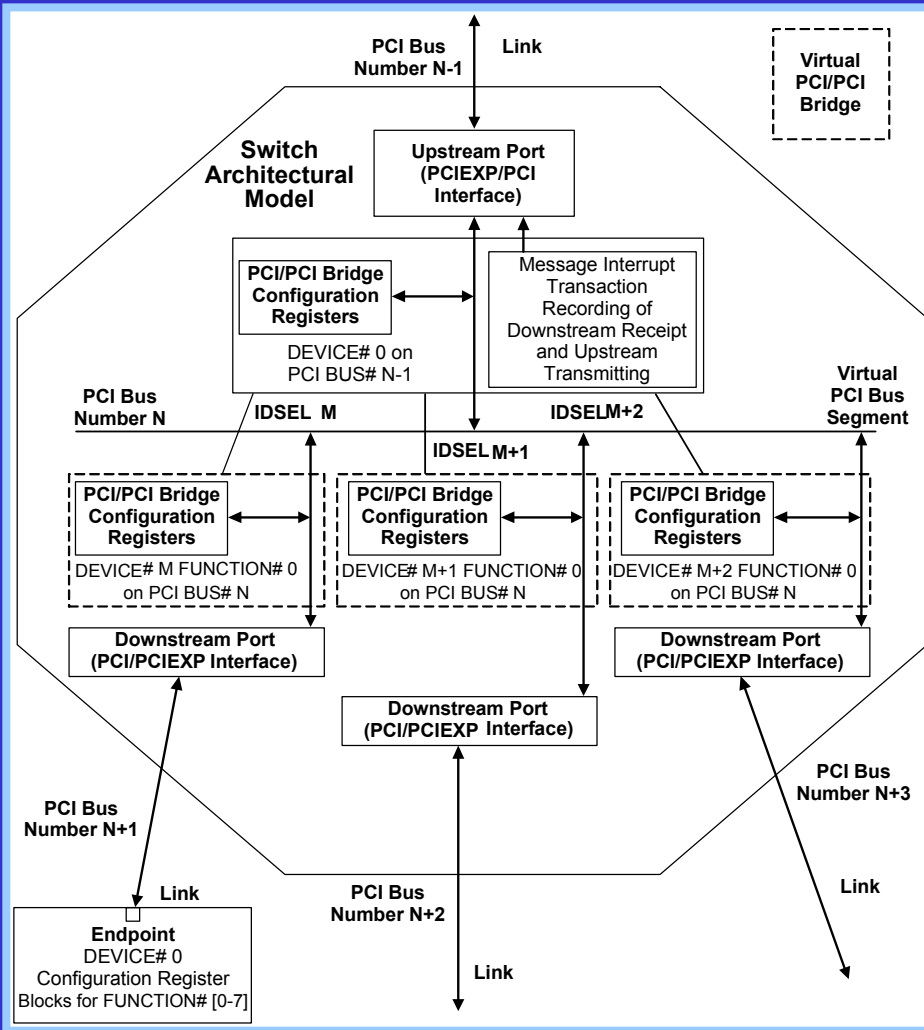
- For a Root Complex executing PCI Express compatible software the size of the RCRB is 4096 Bytes and the access is via the memory address space of the HOST bus segment.
- The exact architecture of the Root Complex is implementation specific. As detailed in Chapter 3 in the book, it is possible to implement the Root Complex differently as follows:
  - Replace all of the multiple **DEVICE#** virtual PCI/PCI Bridges of **FUNCTION# 0** of the downstream ports with a single **DEVICE#** numbered virtual PCI/PCI Bridges with multiple **FUNCTION#s**.
  - Replace some of the multiple **DEVICE#** virtual PCI/PCI Bridges of **FUNCTION# 0** of the downstream ports with a single **DEVICE#** numbered virtual PCI/PCI Bridges with multiple **FUNCTION#s**.





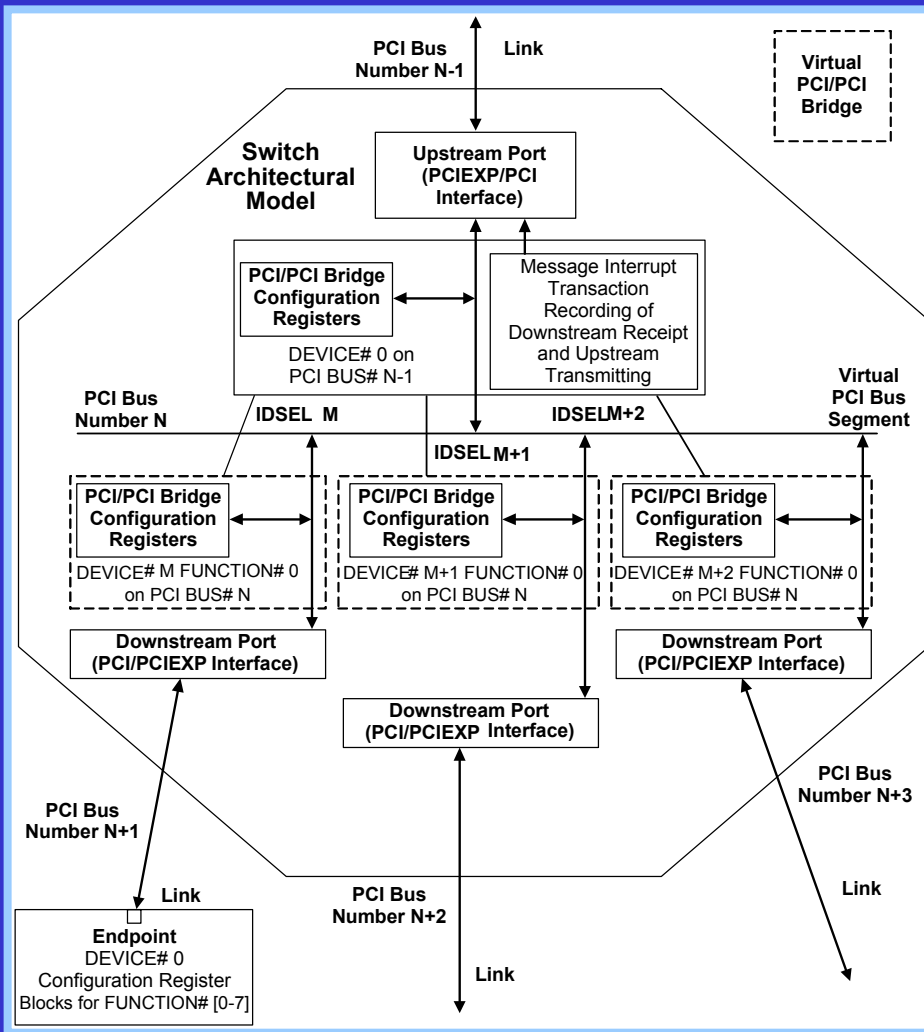
## Switches

- Switches interconnect multiple downstream PCI Express devices on multiple links with a single upstream link. Each link is connected to a port.
- Switches provide fan-out for downstream flowing Transaction Layer Packets (TLPs) and a fan-in for upstream flowing TLPs.
- The typical implementation consists of virtual PCI/PCI Bridges within each port of a switch connected by an internal virtual PCI bus segment.



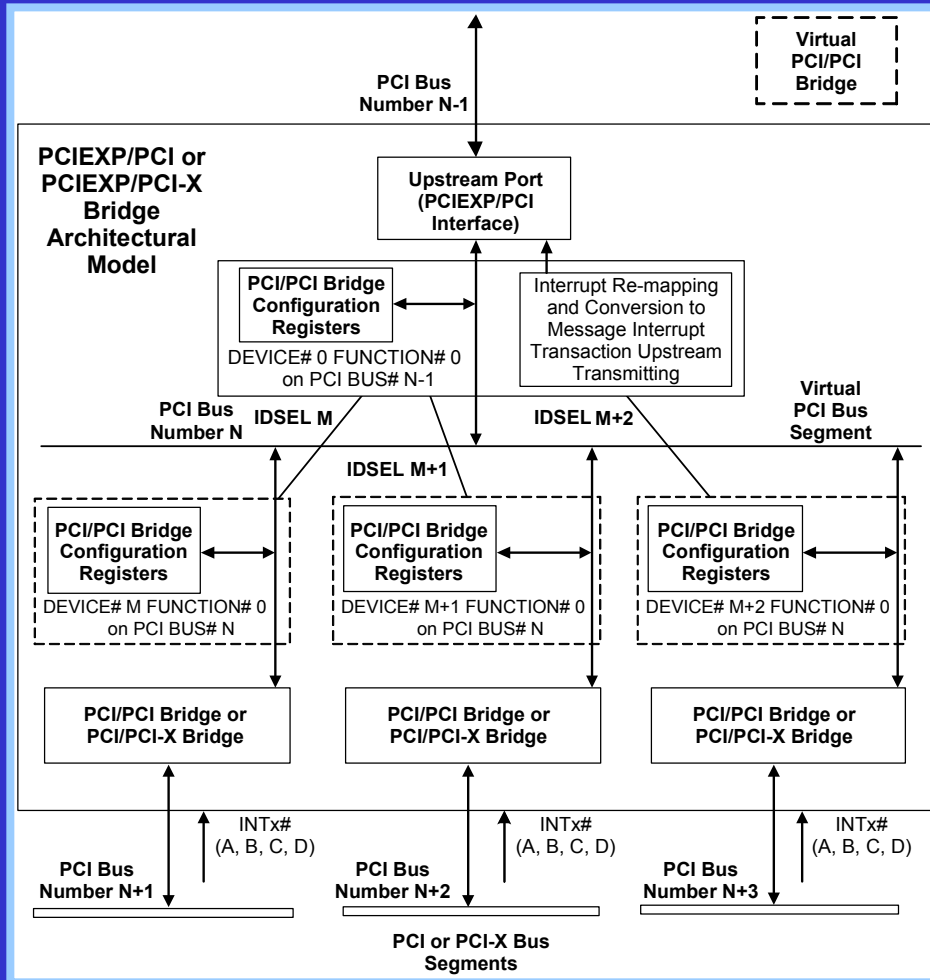
## Switches ...continued

- The exact architecture of the a switch implementation specific. As detailed in Chapter 3 in the book, it is possible to implement a switch differently as follows:
  - Replace all of the multiple DEVICE# virtual PCI/PCI Bridges of FUNCTION# 0 of the downstream ports with a single DEVICE# numbered virtual PCI/PCI Bridges with multiple FUNCTION#s.
  - Replace some of the multiple DEVICE# virtual PCI/PCI Bridges of FUNCTION# 0 of the downstream ports with a single DEVICE# numbered virtual PCI/PCI Bridges with multiple FUNCTION#s..
  - Replace the virtual PCI/PCI Bridge on the upstream port with a direct connection between the upstream link and a local bus segment (replacing the virtual PCI bus segment) that connects to the virtual PCI/PCI Bridges of the downstream ports.



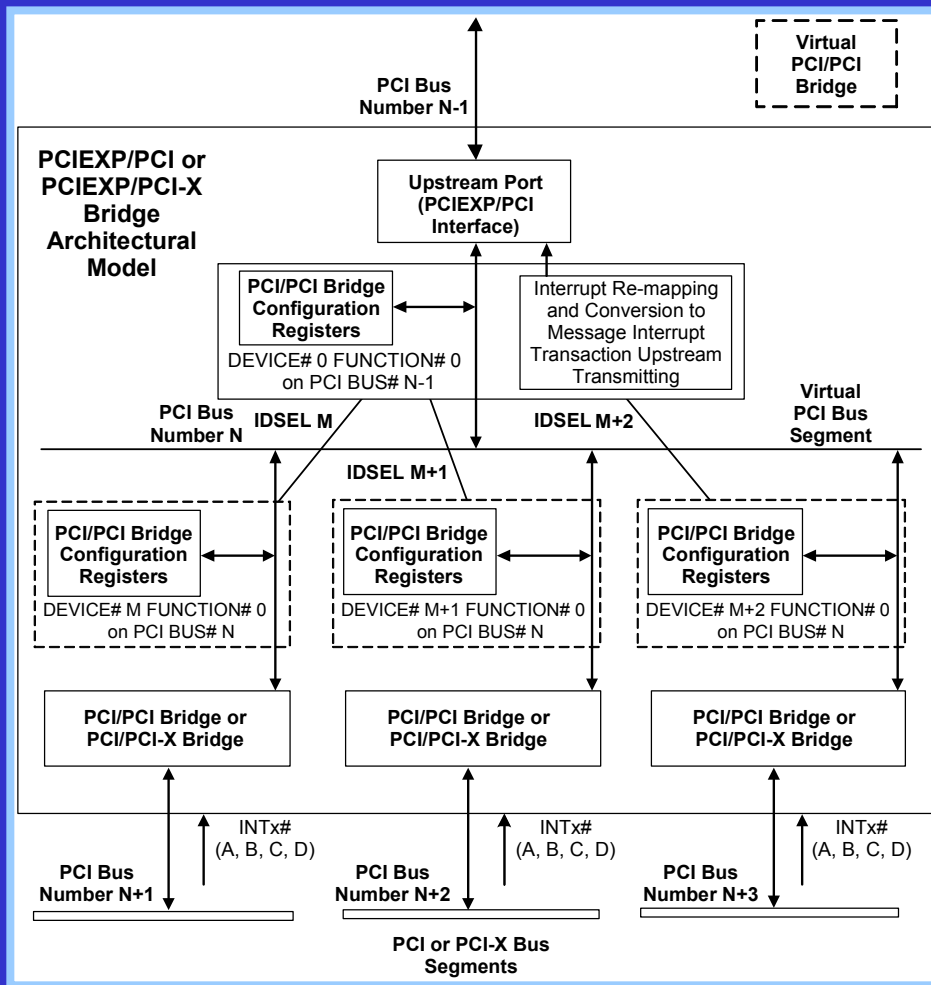
## Endpoints

- Endpoints are PCI Express devices that only connect to a Root Complex or a switch. The connection is via the upstream port of the endpoint to a upstream link which is connected to a downstream port of the Root Complex or a switch.
- Endpoints replace PCI and PCI-X bus masters and targets.
- The endpoints are totally implementation specific, other than the following requirements:
  - Each endpoint is requirement that each contain a configuration register block.
  - The endpoint must be assigned as DEVICE# 0 with 0 to 7 FUNCTION#s.
- There are two types of endpoints: legacy and PCI Express that differ as follows:
  - Legacy endpoints support different PCI Express transactions than the PCI Express endpoints.
  - Legacy endpoints support the lock function, legacy Interrupts, and Message Signaled Interrupt (MSI). Legacy Interrupts implement message transactions to emulate interrupt signal lines. Message Signaled Interrupts (MSI) implement memory write request transactions. PCI Express endpoints do not support the lock function and only implement MSI.



## Bridges

- Bridges interconnect multiple downstream PCI devices on a hierarchy of PCI bus segments with a single upstream link.
- Bridges provide fan-out for downstream flowing Transaction Layer Packets (TLPs) and conversion to PCI bus transactions. Bridges provide fan-in for upstream flowing PCI bus transactions and conversion to TLPs.
- The typical implementation consists of virtual PCI/PCI Bridges within each port connected by an internal virtual PCI bus segment.



## Bridges ... continued

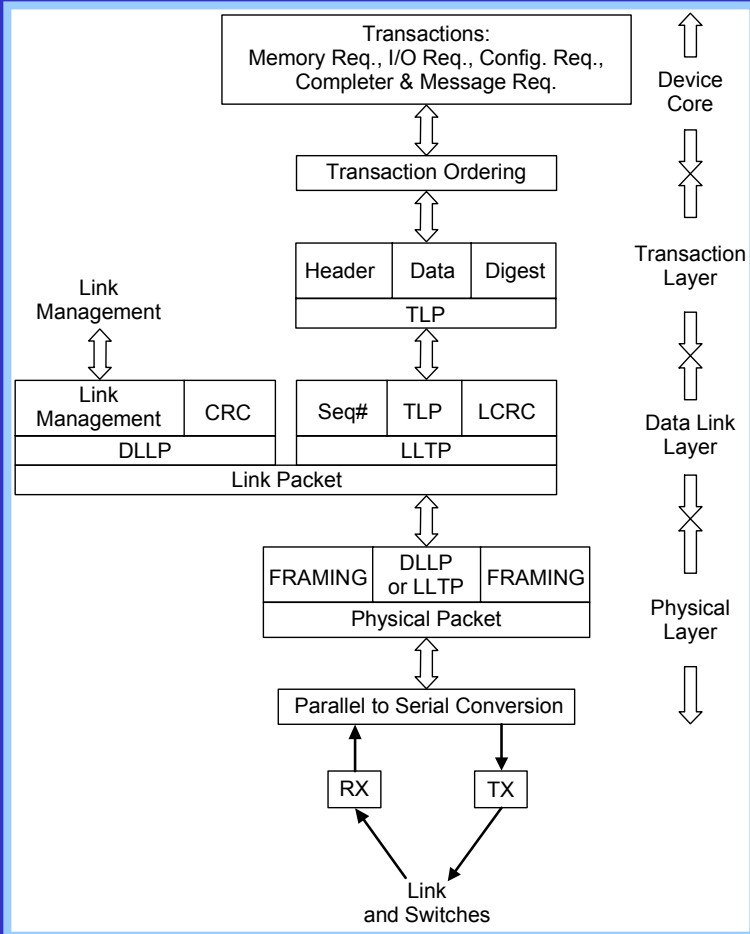
- The exact architecture of the a bridge is implementation specific. As detailed in Chapter 3 in the book, it is possible implement a bridge differently as follows:
  - Replace all of the multiple DEVICE# virtual PCI/PCI Bridges of FUNCTION# 0 of the downstream ports with a single DEVICE# numbered virtual PCI/PCI Bridges with multiple FUNCTION#s.
  - Replace some of the multiple DEVICE# virtual PCI/PCI Bridges of FUNCTION# 0 of the downstream ports with a single DEVICE# numbered virtual PCI/PCI Bridges with multiple FUNCTION#s.
  - Replace the virtual PCI/PCI Bridge on the upstream port with a direct connection between the upstream link and a local bus segment (replacing the virtual PCI bus segment) that connects to the virtual PCI/PCI Bridges of the downstream ports.
- Bridges can also be designed to connect PCI-X bus segments with PCI Express.

## Transition from PCI Express Fabric Architecture to Transactions

- From the previous slides it is obvious that transactions are flowing through a PCI “like” platform within a PCI Express platform. However, there are no PCI “like” bus transactions defined in PCI Express. The following slides discuss the PCI Express transactions with the following considerations.
- PCI transactions are defined for implementation on a bus segment with many parallel signal lines for a bus transaction like format. The PCI Express fabric implements links which do not implement parallel signal lines for bus transaction format. PCI Express transactions have a format that requires only a pair of signal lines in a link with the following considerations:
  - The minimal link consists of a single lane with the addition of lanes to improve link bandwidth. Each lane contains a pair of differentially signal lines per each direction.
  - PCI Express transactions are transmitted across the link the the form of packets.
  - Each PCI Express packet consists of serial symbol stream comprised of a serial bit stream with an integrated reference clock. The serial bits are transmitted onto a differentially driven pair of signal lines
  - PCI Express links can only implement a serial symbol stream for each lane. A multiple lane link simply parses the single stream into multiple parallel streams.

### Transition from PCI Express Fabric Architecture to Transactions ... continued

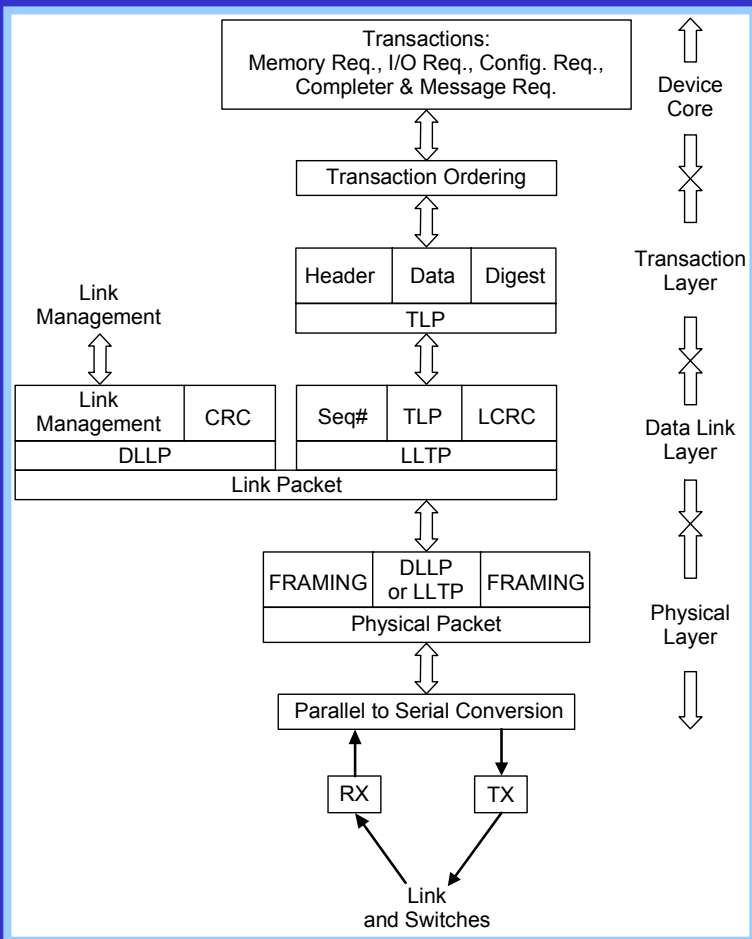
- In a PCI Express platform the **requester source** or **completer source** must convert specific parallel transactions of the PCI Express device core into a serial symbol stream to flow across the links in the PCI Express fabric.
- In a PCI Express platform the **requester destination** or **completer destination** must convert the serial symbol stream that flows across the links in the PCI Express fabric into specific parallel transactions for the PCI Express device core.
- The conversion from the parallel orientation of a PCI Express device core transaction to the serial symbol stream on the link is done in via multiple PCI Express layers. The PCI Express layers transform parallel transactions to serial byte stream and vice versa
- PCI Express layers are between the PCI Express device core and the packets that flow throughout the PCI Express fabric over the links.



## PCI Express Packets Transmitted & Received Between PCI Express Devices

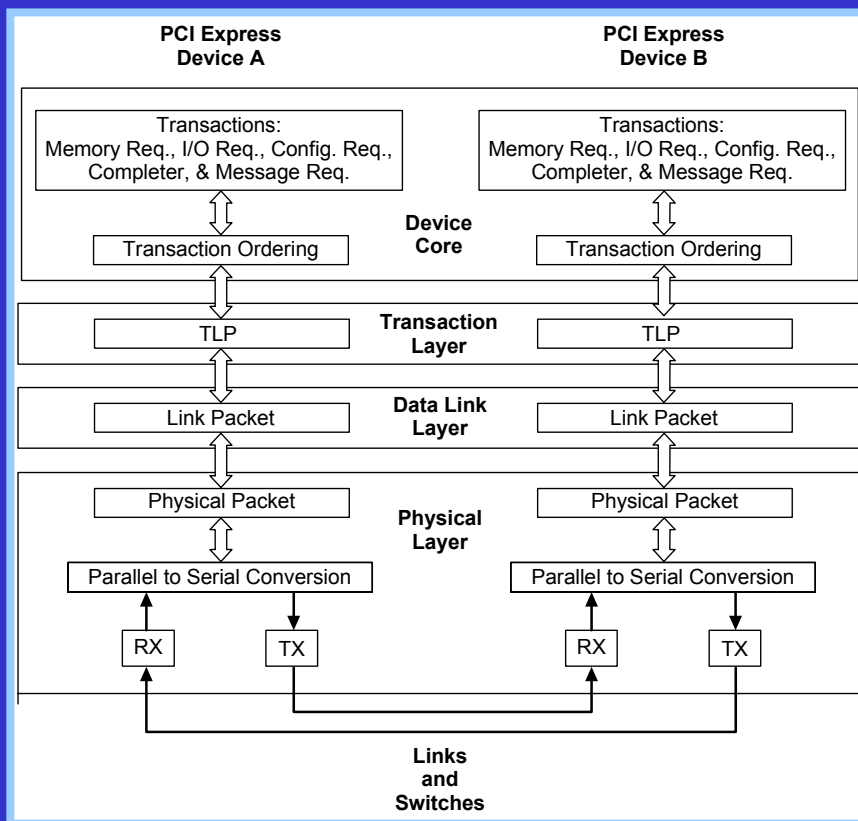
- Each of the PCI Express layer is associated with specific packet as follows:
  - Transaction Layer only defines Transaction Layer Packets (TLPs) that consist of the following:
    - Header field defines the type of transaction: Memory, I/O, configuration, or message; and read or write
    - Data field contains the data to be written for requester transactions and data read for completer transactions.
    - Digest field contains the optional cyclic redundancy checking (ECRC) across the TLP.
  - Data Link Layer defines Link Layer Transaction Packets (LLTPs) and Data Link Layer Packets (DLLPs).
    - LLTPs contain the SEQ# to ensure strict ordering of transmissions across the link and required cyclic redundancy checking (LCRC) for packet integrity.
    - DLLPs contains the link management information and a cyclic redundancy checking (CRC) for packet integrity.





## PCI Express Packets Transmitted & Received Between PCI Express Devices

- Each of the PCI Express layer is associated with specific packet as follows ... continued:
  - Physical Layer only defines Physical Packets that consists of the following:
    - The LLTPs or DLLPs of the Data Link Layer.
    - FRAMING to distinguishes the beginning and end of the symbol stream for each LLTP and each DLLP. FRAMING also distinguishes if the symbol stream is LLTP to DLLP.

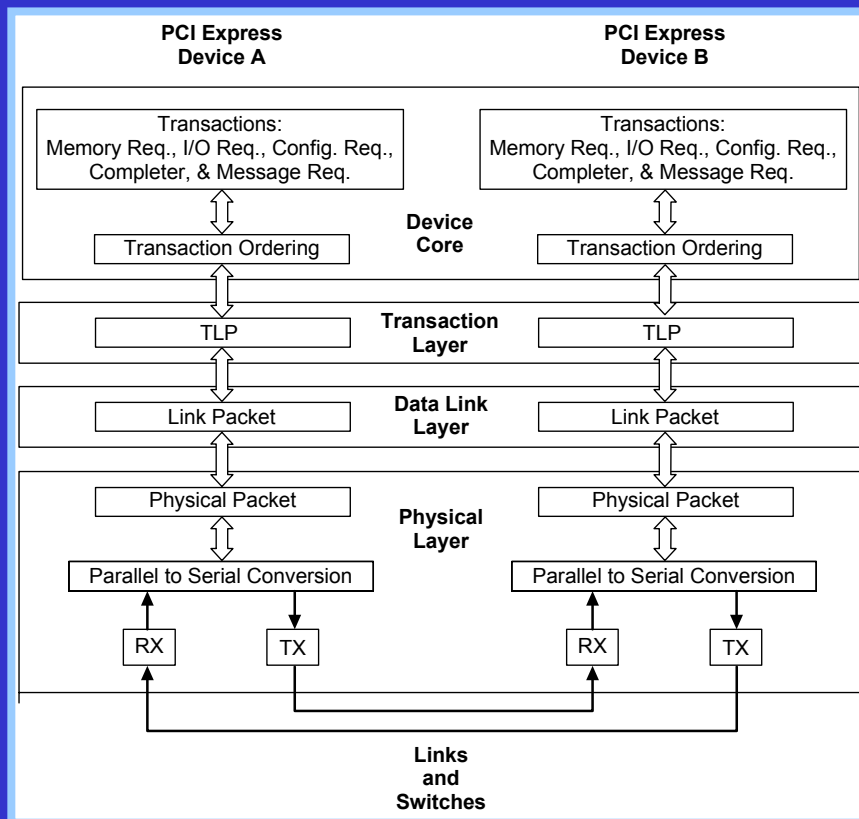


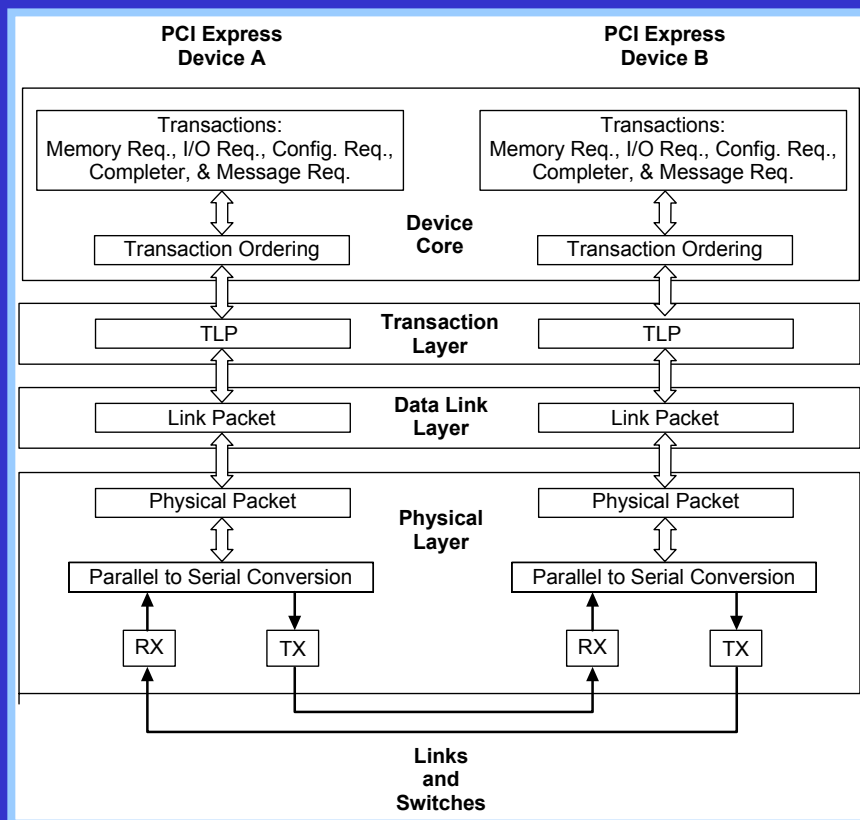
## PCI Express Packets Transmitted & Received Between PCI Express Devices ... continued

- The transmission and reception of PCI Express packets occur on two levels.
- The simple level of interaction is between the PCI Express device that begins the access to another PCI Express device. The PCI Express device accessed responds to the PCI device that requested access.
  - This request defines the requester source and requester destination as the two participants. The response defines the completer source and completer destination as the two participants.
  - If the two participants are on each end of the same link the protocol is simple.
- The complex level of interaction when the two participants are not on the same link. The TLPs exchanged between the two participants must pass through switches.

## PCI Express Packets Transmitted & Received Between PCI Express Devices ... continued

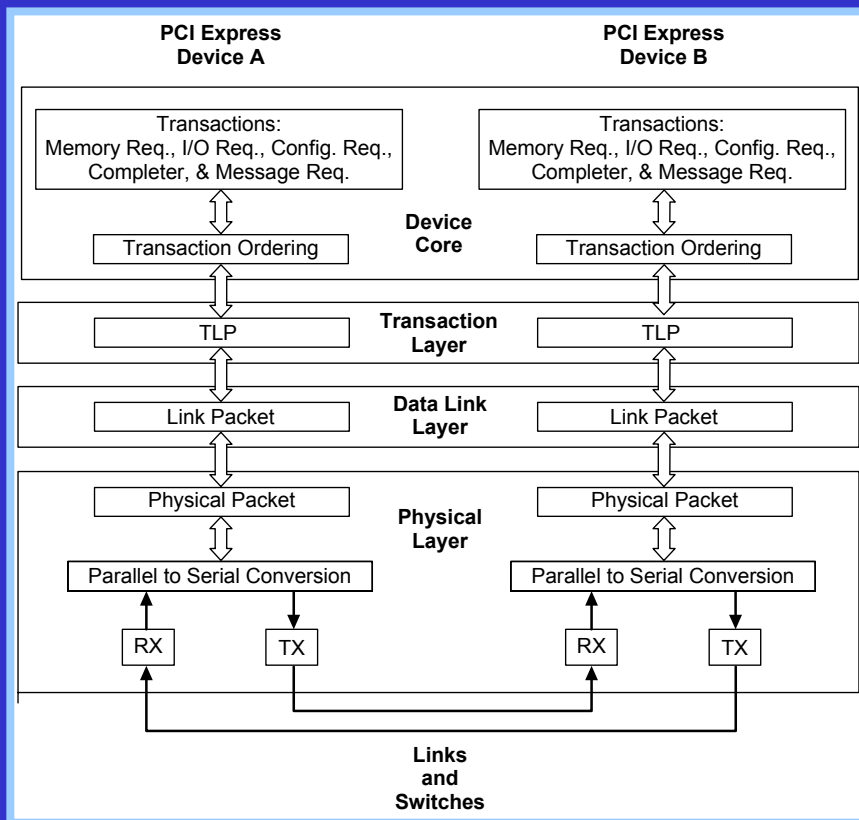
- Simple level of interaction: The following discussion reviews the flow of transactions between PCI Express device cores through layers, and across one link. The discussion follows the Requester/Completer protocol.
- At **requester source** the implementation of the Transaction, Data Link, and Physical Layers is as follows:
  - Transaction Layer: Translates the transactions from the PCI Express device core to a format friendly to the other PCI Express layers in terms of a Transaction Layer Packets (TLPs). TLPs include:
    - Address bits and routing information.
    - Flow Control information in terms of Traffic Class.
    - ECRC optionally provides CRC (cyclic redundancy checking) of the TLPs.
  - Data Link Layer: Encapsulates the TLPs into Link Layer Transaction Packets (LLTPs) which includes the SEQ# and LCRC. The SEQ# preserves strong transaction order over the link. The LCRC retains LLTPs integrity across the link with a CRC of the LLTPs.





## PCI Express Packets Transmitted & Received Between PCI Express Devices ... continued

- At **requester source** the implementation of the Transaction, Data Link, and Physical Layers is as follows:... continued:
  - **Physical Layer:** Encapsulates LLTPs into Physical Packets by executing the following:
    - Conversion of parallel orientation to a serial orientation
    - FRAMING bytes are added to distinguish between specific DLLPs and LLTPs in the serial symbol stream.
    - Application of 8b/10b encoding to integrate a reference clock into the byte and bit stream. The resulting serial bytes and bits stream is defined as a symbol stream.
    - The Physical Layer parses the the symbol stream across multiple lanes within each link.
- **Link:**
  - The Physical Packets transmit from one port to another over the connecting link. The links between the ports of PCI Express devices do not change or modify the Physical Packets.

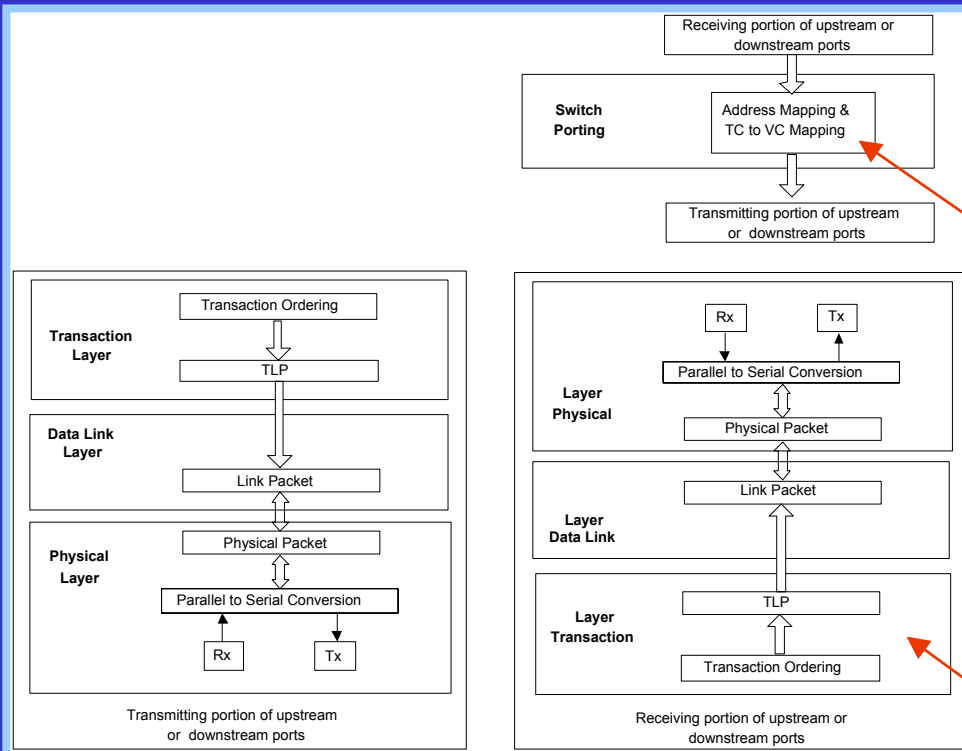


## PCI Express Packets Transmitted & Received Between PCI Express Devices ... continued

- At **requester destination** the implementation of the Transaction, Data Link, and Physical Layers is as follows:
  - Physical Layer: Extract LLTPs from the Physical Packet as follows:
    - The Physical Layer deparses the symbol stream from the link
    - Application of 10b/8b decoding to extract the reference clock from the symbol stream The reference clock define the valid bit periods of the series of bytes in the symbol stream
    - The LLTPs within a series of bytes is distinguished by FRAMING bytes.
    - Conversion of serial orientation to a parallel orientation
  - Data Link Layer: Extracts the TLPs from LLTPs and checks SEQ# for strong ordering and LCRC for the LLTPs' integrity.
  - Transaction Layer: Translates the TLPs from the Data Link Layer to transactions compatible with the PCI Express device core.

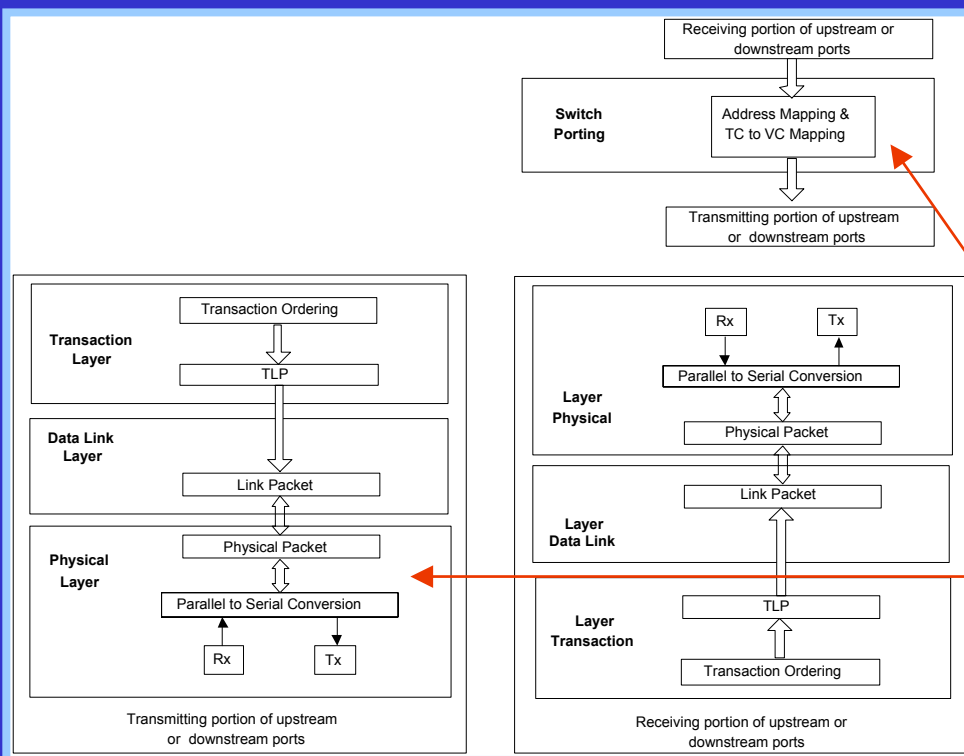
### PCI Express Packets Transmitted & Received Between PCI Express Devices ... continued

- The **requester destination** is also the **completer source**. The requester transaction is processed by the PCI Express device core and a completer transaction will be executed if one is defined for the requester transaction received. The processing of the completer transaction through the layers in the **completer source**, transmits across the link is the same as discussed for the requester transaction.
- The **completer destination** is also the **requester source**. The completer transaction is processed by the layers of the **completer destination** is the same as discussed for the requester transactions. The reception of the completer transaction at the PCI Express device core completes the transaction.



## PCI Express Packets Transmitted & Received Between PCI Express Devices ... Continued

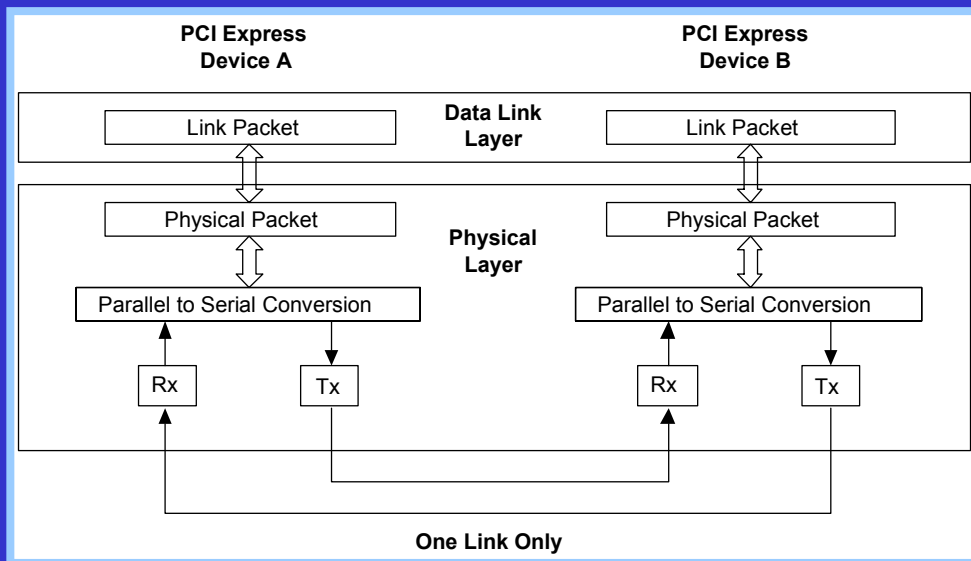
- Complex level of interaction: In not all cases does the two participants exist on the same link as discussed in previous slides. In most cases the two participants are separated by several switches and links.
- The Physical Packet transmitted on each link that contains a LLTP. The Physical Packet and LLTP is only transmitted across one link. The encapsulated TLP is the only entity that is ported through switch and transmitted across links that remain in tact.
- Switches provide a fan-out for TLPs flowing downstream and a fan-in for TLPs flowing upstream.
  - Also defined are TLPs for peer-to-peer transactions that partly flow upstream and downstream with inflection point in the switch.
- The receiving portion of each port of a switch extracts the LLTPs from the Physical Packets.
  - The LLTPs are checked for SEQ# for strong ordering and packet integrity (LCRC).



## PCI Express Packets Transmitted & Received Between PCI Express Devices ... Continued

- Complex level of interaction... continued
- The TLPs are extracted from the LLTPs and are ported internally between ports of the switches.
  - Internally the switches port the TLPs to an output port buffer based on address or routing information within the TLP.
  - The transmission of the TLPs require encapsulation into new LLTPs and Physical Packets.



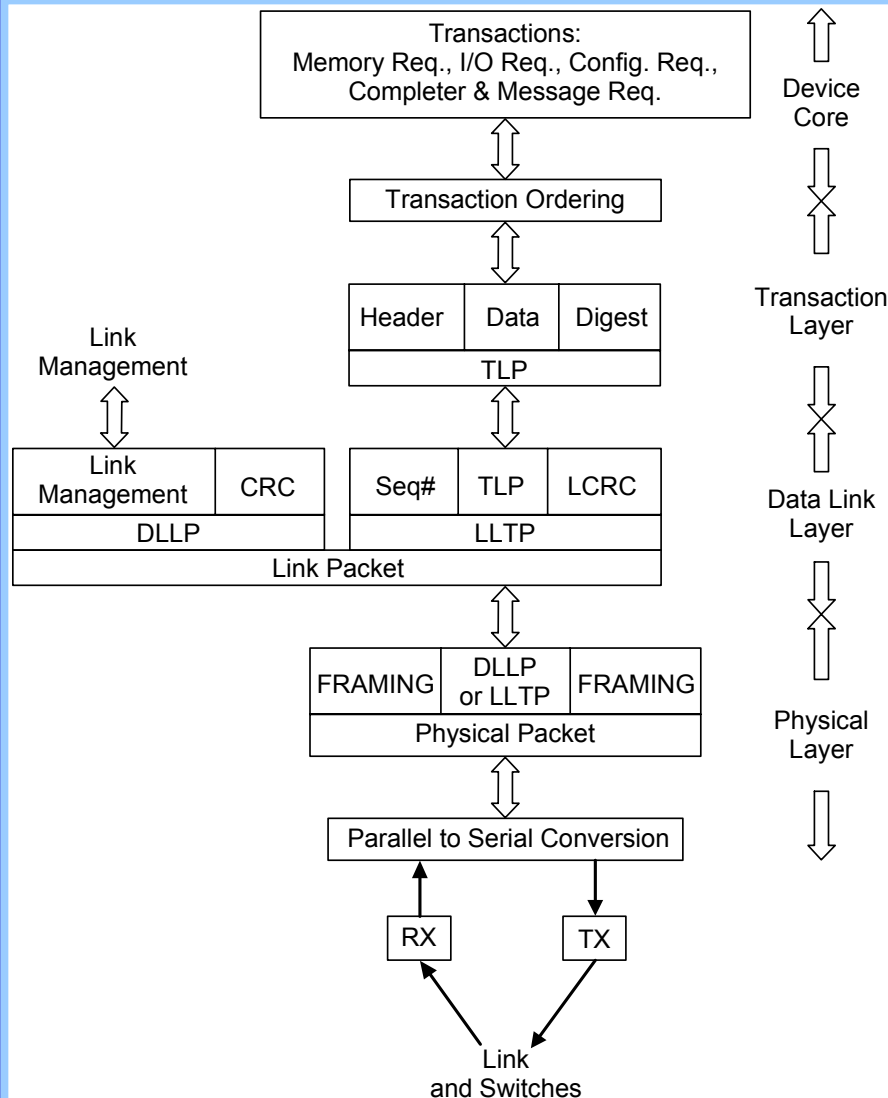


## PCI Express Packets Transmitted & Received Between PCI Express Devices ... Continued

- **Link Management:** In addition supporting LLTPs encapsulating TLPs, the Data Link Layer also encapsulates link management information relative to LLTPs transmissions into Data Link Layer Packets (DLLPs).
- **Transmission and Reception of DLLPs:** Unlike TLPs contained in LLTPs and the link management information is only exchanged between PCI Express devices on the same link.
  - The link management information is never ported through the switch.
- The link management information is implemented only by the Data Link Layer. Consequently, there is no flow and no definition of link management packets between the Transaction Layer and the Data Link Layer. Only between the Physical Layer and the Data Link Layer.

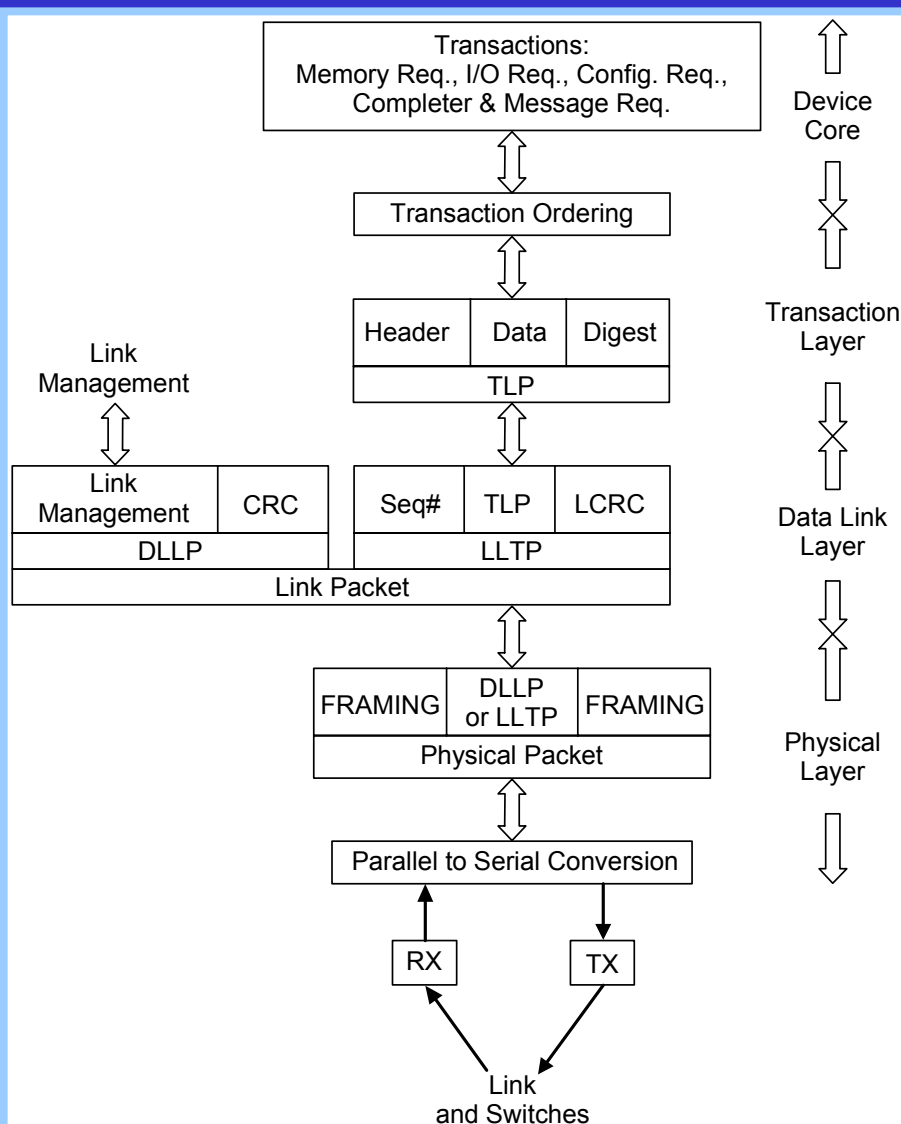
## PCI Express Packets Transmitted & Received Between PCI Express Devices ... Continued

- This link management information is part of the Flow Control protocol and does the following:
  - During Flow Control Initialization (Chapters 11 and 12 of the book) the available buffer space at the receiving port for TLPs contained in LLTPs is determined by exchanging specific Flow Control DLLPs.
  - During normal operation link management tracks the change of available buffer space at the receiving port. The tracking of available buffer space is by specific Flow Control DLLPs that inform the LLTPs' transmitting port on the successful reception at the receiving port.
  - The Flow Control Credits (FCCs) are the values assigned to the different type of transactions and to the amount of associated data. The Data Link Layer uses FCCs as part of Flow Control DLLPs to measure the buffer space available versus the size of TLPs contained in the LLTPs
- Not all DLLPs used for link management are specific to the Flow Control protocol. Other link management DLLPs are used for power management and others are vendor-specific.



## PCI Express Packets Transmitted & Received Between PCI Express Devices ... continued

- All DLLPs are transferred between the ports on a specific link as follows:
  - **Data Link Layer:** At the DLLPs' transmitting port the link management information is encapsulated into the DLLPs with CRC. The CRC retains DLLPs' integrity across the link. The DLLPs are encapsulated into Physical Packets to be transmitted across the link.
  - **Physical Layer:** At the DLLPs' transmitting port the DLLP is encapsulated into Physical Packets by executing the following:
    - Conversion of parallel orientation to a serial bit stream
    - FRAMING bits are added to distinguish between specific DLLPs and LLTPs in the symbol stream.
    - Applies 8b/10b encoding to integrate a clock reference into the bit stream.
    - The resulting serial bit stream is defined as a stream of symbols.
    - The Physical Layer parses the the symbol stream across multiple lanes within each link.
  - The Physical Packets containing DLLPs are immediately transmitted across the link.



## PCI Express Packets Transmitted & Received Between PCI Express Devices ... Continued

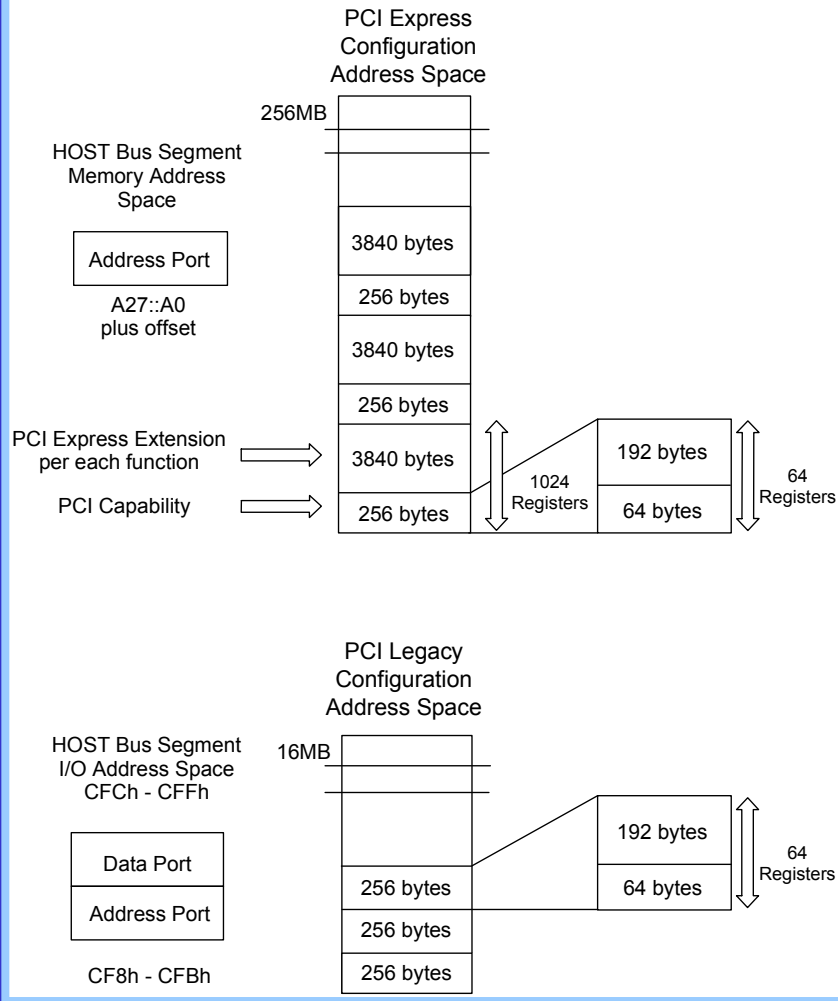
- All DLLPs are transferred between the ports on a specific link as follows: ... continued
  - **Physical Layer:** At the DLLPs' receiving port the DLLPs contained in the Physical Packets are processed as follows:
    - The Physical Layer deparses parallel symbol stream across multiple lanes within each link into a single symbol stream.
    - The Physical Layer extracts the DLLPs from the Physical Packets Applying 10b/8b decoding to extract the clock reference from the symbol stream to reference the valid bit periods. Recover the DLLPs within a symbol stream as distinguished by with FRAMING symbols.
  - **Data Link Layer:** Extracts the link management information from the DLLPs and checks CRC for the DLLPs' integrity. The Data Link Layer implements the link management information.

# Chapter 4

## Addressing, Routing, and Participants

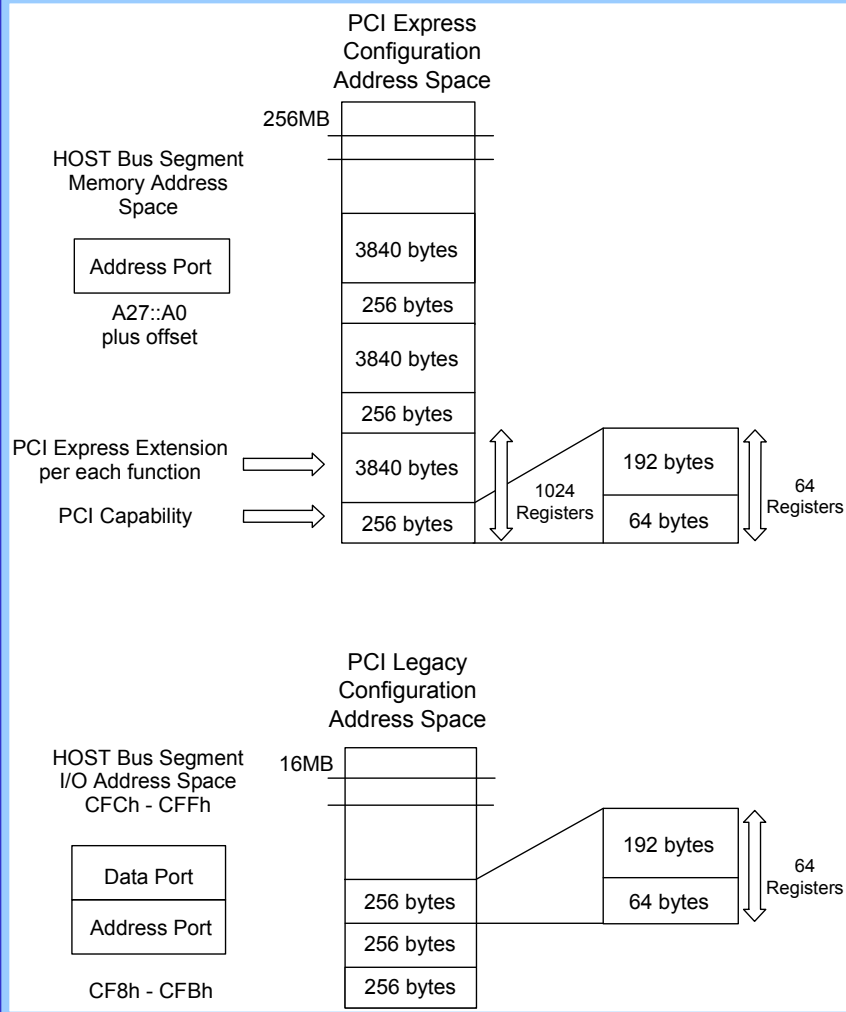
## Addressing

- Memory and I/O address spaces within a PCI Express use the address bits the typical fashion.
- Memory address space
  - Flat across the platform with no alias ... only one PCI Express entity for each address which represents an 8 bit byte.
  - No caching within the PCI Express fabric other than caching of the Platform Memory which is implementation specific.
  - Memory requester transactions with either 32 or 64 address bits defined.
  - The memory address space can be accessed (data can be transferred) as a single 32 data bit access for every one address (lower order two bits always 00b) in one transaction or as multiple 32 data bits accesses (transfers) for one beginning address in one transaction.
- I/O address space
  - Flat across the platform with no alias ... only one PCI Express entity for each address which represents an 8 bit byte
  - I/O requester transactions with 32 address bits defined.
  - Only single accesses of 32 bits is defined for one transaction with one address (lower order two bits always 00b).



## Addressing ..continued

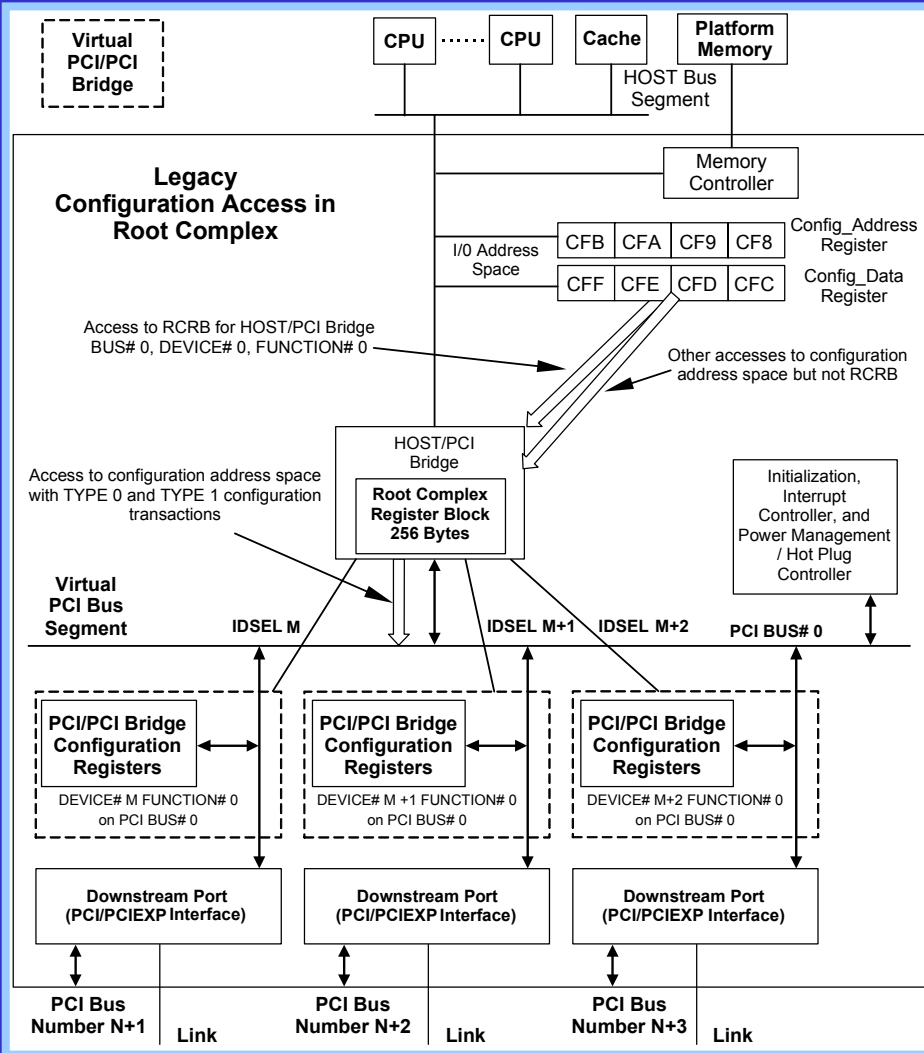
- Configuration address spaces within a PCI Express use the address bits in the typical fashion for the contents within the configuration register blocks. The configuration register blocks are accessed by ID Routing discussed in subsequent slides
  - Flat across the platform with no alias ... only one PCI Express entity for each address which represents an 8 bit byte
  - Configuration address space defines BUS#, DEVICE#, and FUNCTION#
    - BUS# defines one of 256 possible "bus segments" Each link is assigned a BUS#, each virtual PCI bus segment is assigned a BUS#, and each PCI or PCI-X bus segment downstream of a bridge is assigned a BUS#.
    - DEVICE# defines one of 32 virtual PCI Express devices in the form of virtual PCI devices internal to a Root Complex, switch, or bridge. An endpoint can only be assigned DEVICE# 0.
    - Each specific DEVICE# can have one to eight FUNCTION#s.
    - Each function defines one configuration register block.



## Addressing ... continued

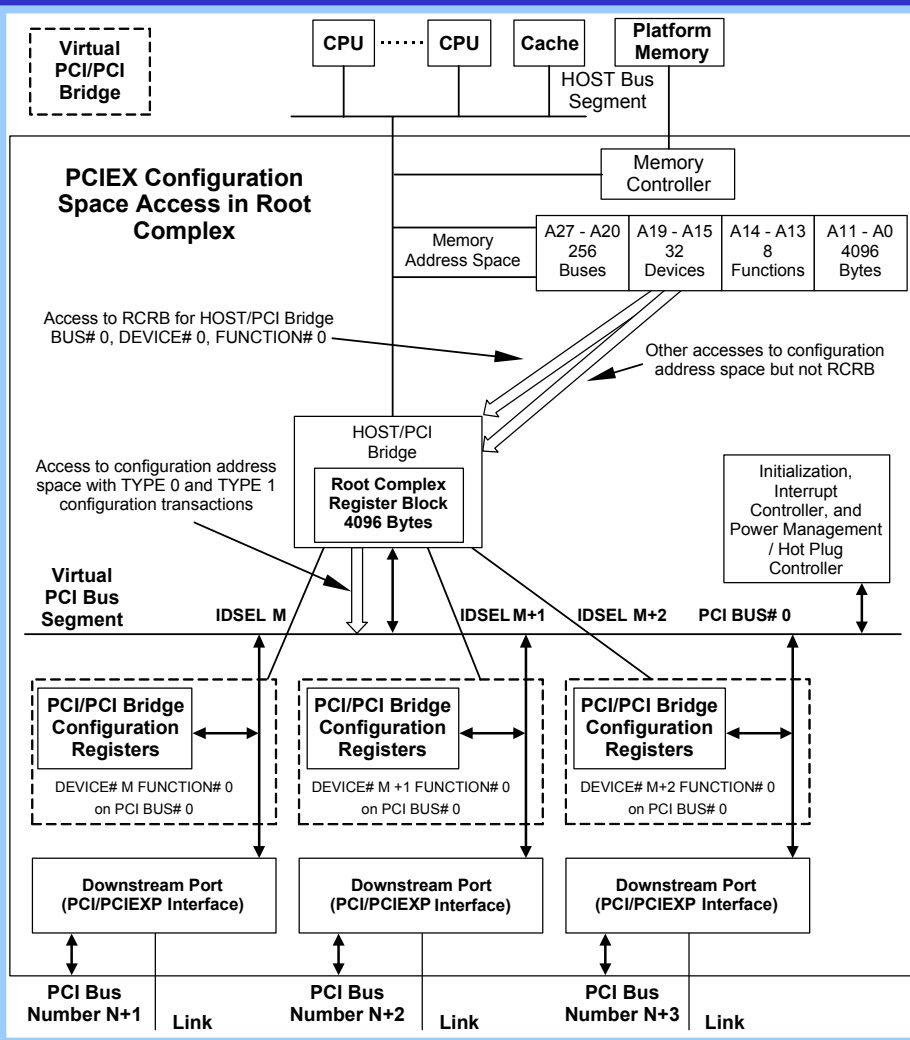
- Configuration address spaces within a PCI Express device uses the address bits in the typical fashion for the contents within the configuration register blocks. The configuration register blocks are accessed by ID Routing discussed in subsequent slides .. continued
  - Configuration address space also defines REGISTER#
    - The 8-bit bytes are configured into 32 bit registers with the configuration registers block of each function.
    - For a PCI compatible software (legacy software) 64 registers are defined in the configuration register block.
    - For a PCI Express compatible software 1024 registers are defined in the configuration register block.





## Addressing ... continued

- Root Complex Considerations
  - The basic Root Complex is implemented with a virtual HOST/PCI Bridge and individual virtual PCI/PCI Bridges for each downstream port and are the PCI Express devices assigned configuration register blocks. One configuration address block for each FUNCTION#.
  - The configuration register block for the virtual HOST/PCI Bridge is called the Root Complex Register Block (RCRB).
  - Accessing configuration address space within a Root Complex is done in two ways:
    - For PCI compatible software (legacy software) the access to the RCRB and the configuration register blocks associated with the virtual PCI/PCI Bridges of the downstream ports is via the I/O address space of the HOST bus segment.



## Addressing ..continued

- Root Complex Considerations .. continued
  - Accessing configuration address space within a Root Complex is done in two ways .. continued:
    - For PCI Express compatible software the access to the RCRB and the configuration register blocks associated with the virtual PCI/PCI Bridges of the downstream ports is via the memory address space of the HOST bus segment.

## Routing

- Routing is used by some PCI Express transactions
  - As discussed in the earlier slides, the memory, I/O, and contents of the configuration register blocks within a PCI Express use the address bits in the the typical fashion. Per the Requester/Completer, the requester transaction use the address bits; however, any associated completer transactions use ID Routing instead of address bits.
  - As discussed in the earlier slides, the requester transactions used to access the configuration register blocks also use ID Routing.
- ID Routing consists using BUS#, DEVICE# and FUNCTION# within the Header field of the configuration requester transactions and all completer transactions. The combination of BUS#, DEVICE# and FUNCTION# are defined as CONFIGURATION, REQUESTER, COMPLETER and VENDER-DEFINED IDs.
  - CONFIGURATION and VENDER-DEFINED ID is used by the **requester source** to provide routing information for configuration and message vender-defined requester transactions, respectively.
  - REQUESTER ID is used by the **completer source** to provide routing information for completer transactions.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
7      BUS NUMBER      0								4      DEVICE #      0					2      FUNC#      0					
15								REQUESTER ID HDW								00		

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0			
7      BUS NUMBER      0								4      DEVICE #      0					2      FUNC# 0					
15								COMPLETER ID HDW								00		

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		1	0	2r FIELD 0			R	2 TC 0		R	R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG 0								7 MESSAGE CODE 0							
63 ADDRESS HDW								48							
47 ADDRESS HDW								32							
31 ADDRESS HDW								16							
15 ADDRESS HDW								00							

## Routing ... continued

- Implied Routing consist of the r field in the Header field of the message requester transactions. The r field information routes the message baseline requester transaction and vender-defined requester transactions. Not every Implied Routing can be implemented by every PCI Express device.
  - r = 000b:** The destination of message requester transaction packets is the Root Complex from any one of the sources defined in tables.
  - r = 011b:** The source of message requester transaction packets is the Root Complex and the destination is all PCI Express devices listed in the tables (broadcast to multiple destinations).
  - r = 100b:** The source and destination are on the same link. The associated message requester transaction packets are not ported through a switch.
  - r = 101b:** Gathered from several sources by switches and sent to Root Complex, this routing only applies to message PME\_TO\_ACK transactions.
- RID Routing is the acronym for Route Identifier and is implemented only by the message advanced switching requester transactions.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	T	R	2	TC		0	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG 0								3 LAST BE 0				3 FIRST BE 0			
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
R	R	R	R	EXT, REGISTER 3 ADDRESS 0				REGISTER ADDRESS 5 0					R	R	

## TYPE 0 and 1 Configuration Requester Transaction Routing

- Configuration requester transactions are unique with two types defined in Header field.
  - When a TYPE 1 configuration requester transaction is received by the upstream port of a switch or bridge, the switch or bridge must port the associated TLP to the proper downstream port per the ID Routing information.
  - When the downstream port of the Root Complex or a switch has determined that the BUS# it is connected to equals the BUS# in a TYPE1 configuration requester transaction ported from the upstream port the following occurs:
    - The downstream port must convert the TYPE 1 configuration requester transaction to a TYPE 0 and then transmit downstream.
  - When TYPE 0 configuration requester transaction is received by the upstream port of a switch, bridge, or endpoint, the access is to a configuration register block internal to the switch, bridge, or endpoint.
  - The above concepts apply to the virtual PCI/PCI Bridges and virtual PCI bus segment internal to the Root Complex, switches, and bridges.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	L	R	2	TC 0		R	R	R	R
T	E	ATTR 1 0	R	R	9 LENGTH 0										
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG# 0								3 LAST BE 0				3 FIRST BE 0			
31 ADDRESS								16							
15 ADDRESS								02 R R							

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	T	R	2	TC	0	R	R	R	R
T D	E P	ATTR 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0							4	DEVICE # 0				2 FUNC# 0			
2 0		STATUS		B C M	11 BYTE COUNT 00										
7 BUS NUMBER 0							4	DEVICE # 0				2 FUNC# 0			
7 TAG # 0							R	6 LOWER ADDRESS 0							

## TAG# and Transaction ID

- The Requester/Completer protocol requires the **requester destination** for certain requester transactions to become a **completer source** for an associated completer transaction packet.
- At the **requester source** it is possible to have transmitted many requester transaction packets prior to receiving any associated completer transaction packets.
- In order for the **requester source** to determine which completer transaction packets received are responses to which requester transaction packets transmitted, the **requester source** includes a unique TAG# in the Header field of each requester transaction packets.
- The **completer source** returns the TAG# received in the requester transaction packets in the complete transaction packets it transmits.

# The Complete PCI Express Reference Topic Group 2 Tutorial

Disclaimer: Intel, Research Tech Inc. and the authors of this tutorial make no warranty for the accuracy or use of the information. No direct or indirect liability is assumed and the right to change any information without notice is retained.

## Design Tools for PCI Express

The PCI Express specification is not organized by clear design topics, does not provide sufficient details to easily master PCI Express, and does not provide in depth illustrations and flowcharts to assist in designing components. Intel recognized that simply summarizing or re-wording the specification as typically done in the other design tools in the industry is insufficient ... more extensive information is are needed.

To provide designers with PCI Express design and implementation information that is easy to master and detailed enough to assist in correct design, two design tools are available:

Six Detailed Tutorials and a new and exhaustively detailed design book.

These design tools focus on Six Quick and Easy Topic Groups which simplify the mastery of PCI Express. They save a designer weeks of trying to unravel the specification and provide the assurance of correct design implementation the first time.



## Design Tools for PCI Express

### The “Book”

The primary design tool that provides total design mastery is *The Complete PCI Express Reference* book written by Edward Solari and Brad Congdon and published by Intel ... referred to as the “Book”.

The Book provides the complete and extensive narrative of detailed figures (over 250), detailed design flow charts, and exhaustive tables for the complete understanding and design assistance in over 1000 pages. The Book can be ordered at [www.amazon.com](http://www.amazon.com) ... ISBN # 0971786194.

### Detailed Tutorials

Six free **Detailed Tutorials** ... One self paced tutorial for each of the Six Quick and Easy Topic Groups. Each introduces PCI Express information with a narrative that complements detailed figures, flow charts, and tables for each specific Topic Group from the Book. The six free **Detailed Tutorials** are available at [www.intel.com/intelpress/pciexpresscomplete](http://www.intel.com/intelpress/pciexpresscomplete).

### This Detailed Tutorial is of Topic Group 2

Detailed Tutorial: *Packets’ and Layers’ Specifics and Errors*

References in the Book: *Chapters 5 to 9*

## PCI Express in Six Topic Groups

### Topic Group 1

**Tutgroup1: *Platform Architecture and Accessing of Resources within Architecture***

**References in the Book: *Chapters 1 to 4***

### Topic Group 2

**Tutgroup2 : *Packets' and Layers' Specifics and Errors***

**References in the Book: *Chapters 5 to 9***

### Topic Group 3

**Tutgroup3 : *Transaction Ordering and Flow Control Part 1 and 2 Protocols***

**References in the Book: *Chapters 10 to 12***

### Topic Group 4

**Tutgroup4 : *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

**References in the Book: *Chapters 13 to 17***

### Topic Group 5

**Tutgroup5 : *Other Hardware Topics***

**References in the Book: *Chapters 18 to 21***

### Topic Group 6

**Tutgroup6 : *Software Considerations***

**References in the Book: *Chapters 22 to 24***

# Packets' and Layers' Specifics and Errors

## Chapters 5 to 9

### Topic Group 2

The interaction between PCI Express device cores and the interconnecting links is done on three layers.

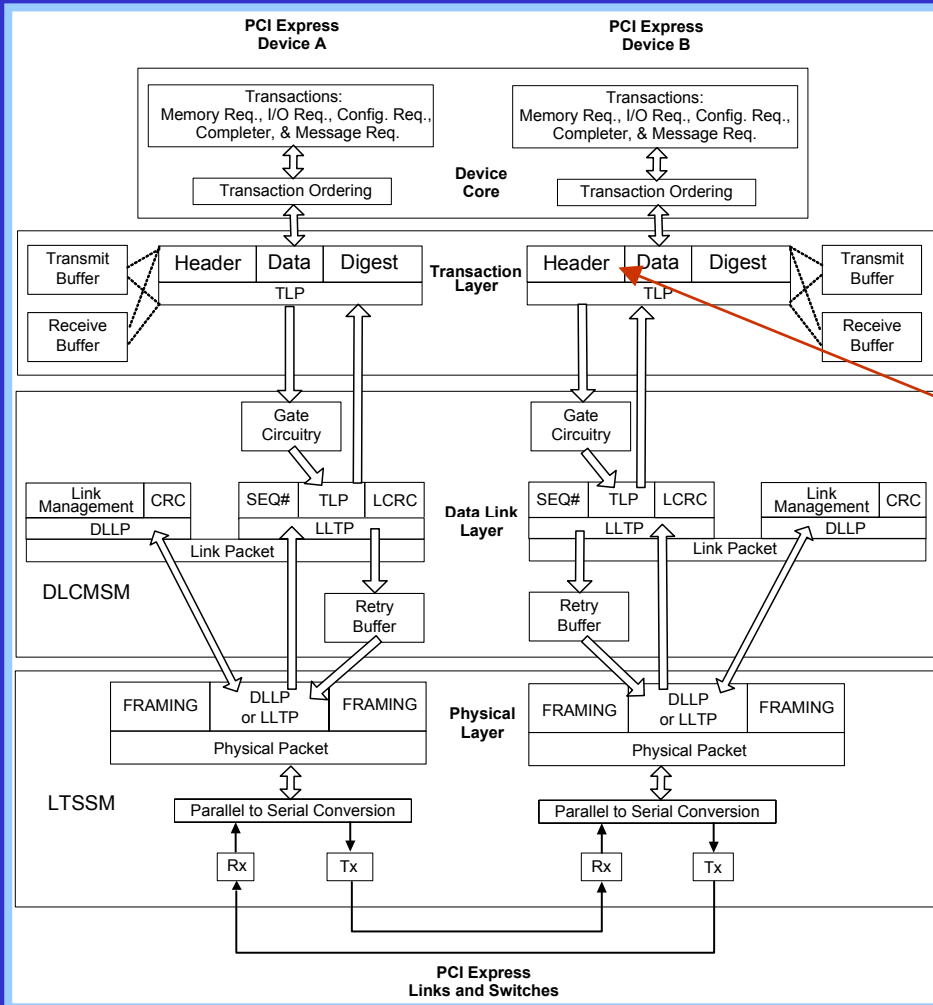
**Summary:** Each of the three layers is implemented in each PCI Express device and each layer has a unique packet. The conversion and transmission of packets may incur errors that must be checked and reported.

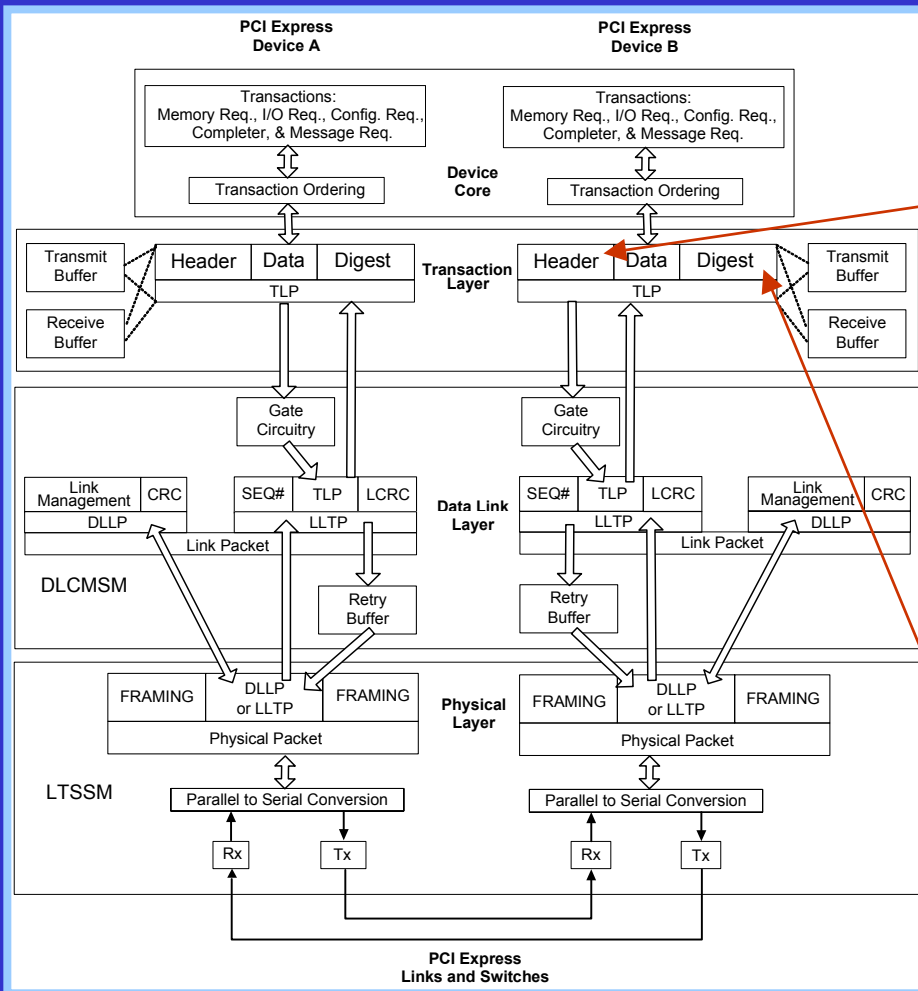
# Chapters 5 & 6

## Transaction Packets & Layer

## Transaction Layer

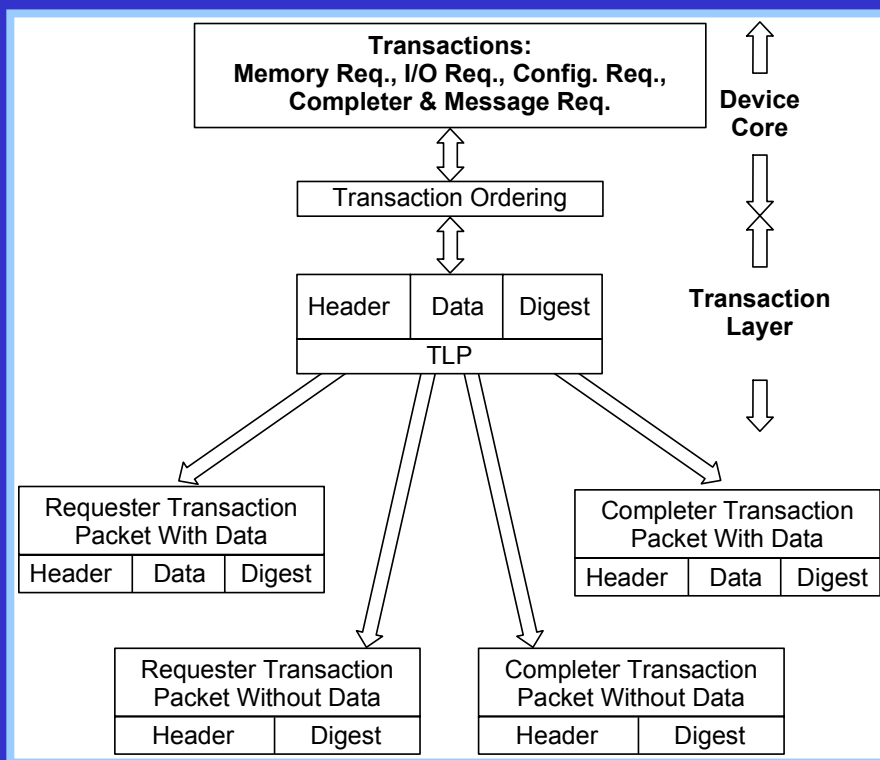
- As discussed in earlier tutorials there are three PCI Express layers in each PCI Express device that provides an interface between the PCI Express core and the PCI Express link.
- The Transaction Layer provides the interface between the PCI Express device core and the Data Link Layer with the purpose to implement elements basic to the PCI Express transaction.
- These elements are dependent on the address space being accessed and are included in the Header field of the Transaction Layer Packet (TLP) as following:
  - Conversion from the HOST bus segment's address spaces to the address and routing format of PCI Express and vice versa.
  - Encoding/decoding the the type of transaction and the implementation of EP bit that indicates if the data is sourced with an error.
  - The implementation of ATTRIBUTE bits that provides information on ordering and snooping.
  - The implementation of Traffic Class numbers to provide information for the Flow Control protocol.





## Transaction Layer ... continued

- These elements are dependent on the address space being accessed and are included in the Header field of the Transaction Layer Packet (TLP) as following: .. continued
  - The implementation of Traffic Class numbers to provide information for the Flow Control protocol.
  - The implementation of LENGTH and BYTE ENABLEs to identify the number of DWORDs and the valid bytes of the first and last DWORD.
  - The BYTE COUNT for the completer transaction.
  - The Requester and Completer ID to provide information or the source of the requester or completer transaction.
  - The TAG# to provide a connection between a requester transaction and a completer transaction.
- The addition (optional) of the cyclic redundancy check (ECRC) information contained in the Digest field of the TLP.



## Transaction Layer Packets (TLPs)

- As discussed in the first tutorial each PCI Express layer has an associated packet. The Transaction Layer Packet (TLP) represents requester and completer transaction packets.
- All TLPs are structured with a required Header field, an Data field when applicable, and an optional Digest field.
  - The Header field contains all of the unique addressing and routing information for the TLP.
  - The Data field is required in all requester transaction packets for writing data and all completer transactions associated with read requester transactions
    - If the data can not be successfully read, the completer transaction does not contain a Data field
  - The Digest field is optional and contains the cyclic redundancy check information for the TLP called the ECRC.
- Subsequent slides will focus on two aspects of the TLP: Single Versus Multiple Byte Accesses and the format of the Header field.

## Major Distinctions Between the Different Types of TLPs

- Memory write requester transaction packets have no associated completer transaction packets.
- Each Memory read requester transaction packet has a “set” of associated completer transaction packets (“set” to be discussed in subsequent slide).
- Each I/O or configuration read and write requester transaction packet has a single associated completer transaction packet. For a write the completer transaction packet it is confirming completion, for a read it is confirming completion and is providing the read DWORD.
- Message requester transaction packets have no associated completer transaction packets.
- The message requester transaction packets actually consists of three different types: baseline, vender-defined, and advanced switching. Vender-defined is implementation specific and advanced switching is still under development.
- The message baseline requester transaction packets support the following:
  - Interrupt: Provides emulation of PCI defined INTx# signal lines.
  - Error: Reports errors to the Root Complex.
  - Hot Plug Protocol: A slot can optionally support the insertion and removal of an add-in card while main power is applied to the PCI Express platform.
  - Power Management: Used to lower the power level of links and PCI Express devices and to prepare all or part of a PCI Express platform to be powered-off or placed into sleep.
  - Slot Power: Permits power limits to be applied individually to each slot
  - Lock: Provides emulation of PCI defined LOCK# signal line.



## TLPs for Memory and I/O Address Spaces

- All memory and I/O TLPs have certain common bit fields in the Header field related to defining the Data accessed as exemplified for requester and completer transaction packets .
- Requester:
  - FMT: Defines the number of DWORD in Header field.
  - TYPE: Defines memory, I/O, configuration, and message; and read or write requester transaction versus a completer transaction. Also defines if part of Lock function (L).
  - TC (Traffic Class): Defines the group of TLPs to be associated with per Flow Control protocol
  - TD: Defines that a Data field is attached.
  - EP: Defines the Data field has errors.
  - ATTRIBUTE: Defines the application of transaction ordering and snooping to the TLP.
  - LENGTH: The DWORDs in the Data field.
  - BUS#, DEVICE#, and FUNCTION#: Defined for ID Routing information in configuration requester transaction packets. Defined as Requester ID and used in the associated completer transaction.
  - TAG#: Provides a unique number to associate a specific requester transaction with a completer transaction.
  - FIRST and LAST BE [3::0]: Bytes enables of the first and last DEWORD (Discussed in subsequent slides).
  - ADDRESS: The number of address bits varies between memory (32 and 64 ) and I/O (32 defined).

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0				
R	FMT 1 0		4	TYPE		1	L	R	2	TC 0		R	R	R	R				
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0													
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0							
7 TAG 0								3 LAST BE 0				3 FIRST BE 0							
31								ADDRESS								16			
15								ADDRESS								02 R R			

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
R	FM 1		4		TYPE		1	L	R	2		TC	0	R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0											
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0					
STATU 2		B C M		11 BYTE COUNT 00													
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0					
7 TAG # 0								R	6 LOWER ADDRESS 0								

## TLPs for Memory and I/O Address Spaces ... continued

- All memory and I/O TLPs have certain common bit fields in the Header field related to defining the Data accessed as exemplified here for requester and completer transaction packets: ... continued
- Completer:
  - FMT: Defines the number of DWORDs in the Header field.
  - TYPE: Defines a completer transaction versus a requester transaction and if associated with the Lock function (L)
  - TC (Traffic Class): Defines the group of TLPs to be associated with per Flow Control protocol.
  - TD: Defines that a Data field is attached.
  - EP: Defines the Data field has errors.
  - ATTRIBUTE: Defines the application of transaction ordering and snooping to the TLP.
  - LENGTH: The number of DWORDs in the Data field.
  - BUS#, DEVICE#, and FUNCTION#: One set is Requester ID used to indicate the destination of the completer transaction per ID Routing protocol. The other set is the Completer ID used to indicate the source of the completer transaction.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	L	R	2	TC		0	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0							4 DEVICE # 0				2 FUNC# 0				
STATUS 2 0		B C M	11 BYTE COUNT 00												
7 BUS NUMBER 0							4 DEVICE # 0				2 FUNC# 0				
7 TAG # 0							R	6 LOWER ADDRESS 0							

## TLPs for Memory and I/O Address Spaces ... continued

- Completer: ... continued
  - STATUS: Defines if completer transaction represent a successful or unsuccessful access.
  - BCM bit supports a unique PCI-X protocol requirement.
  - BYTE COUNT: Number of bytes of data to be returned in total.
  - TAG#: Provides a unique number to associate a specific requester transaction with a completer transaction.
  - LOWER ADDRESS: Defines a portion of the lower base address base for data read.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
R	FMT 1 0		4		TYPE		1	T	R	2		TC	0	R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0											
7 BUS NUMBER 0								4 DEVICE # 0					2 FUNC# 0				
7 TAG 0								3 LAST BE 0					3 FIRST BE 0				
7 BUS NUMBER 0								4 DEVICE # 0					2 FUNC# 0				
R	R	R	R	EXT, REGISTER 3 ADDRESS 0				REGISTER ADDRESS 5 0						R	R		

## TLPs for Configuration Address Space

- Configuration TLPs have certain common bit fields in the Header field common to the memory and I/O address space and some additional bits defined as follows for requester and completer transaction packets.
- Requester:
  - FMT: Defines the number of DWORD in Header field.
  - TYPE: Defines configuration versus memory, I/O, and message. Defines read or write, and Type of configuration (T).
  - TC (Traffic Class): Defines the group of TLPs to be associated with per Flow Control protocol.
  - TD: Defines that a Data field is attached.
  - EP: Defines the Data field has errors.
  - ATTRIBUTE: Defines the application of transaction ordering and snooping of the TLP.
  - LENGTH: The number of DWORDs in the Data field.
  - BUS#, DEVICE#, and FUNCTION#: One set is used to indicate the destination of the configuration requester transaction. The other set is the Requester ID used to indicate the source of the configuration requester transaction.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	T	R	2	TC 0		R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG 0								3 LAST BE 0				3 FIRST BE 0			
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
R	R	R	R	EXT, REGISTER 3 ADDRESS 0				REGISTER ADDRESS 5 0					R	R	

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	L	R	2	TC		0	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
STATUS 2 0		B C M	11 BYTE COUNT 00												
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG # 0								R	6 LOWER ADDRESS 0						

## TLPs for Configuration Address Space

- Configuration TLPs have certain common bit fields in the Header field common to the memory and I/O address space and some additional bits defined as follows for requester and completer transaction packets.
- Requester: ... continued
  - TAG#: Provides a unique number to associate a specific requester transaction with a completer transaction.
  - FIRST and LAST BE [3::0]: Bytes enables of the first and last DWORD (Discussed in subsequent slides).
  - EXT REG. ADDRESS and REGISTER ADDRESS: Defined for configuration requester transaction packet to select a DWORD within the configuration register block.
- Completer: Basic completer transaction is the same for all address spaces and is the same as defined in a previous slides for the memory and I/O requester transactions.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE			0	R	2	TC	0	R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG 0								7 MESSAGE CODE 0							

## TLPs for Message Address Space

- Message TLPs have certain bit fields in the Header field common to the memory and I/O address space and some additional bits defined as follows for requester and completer transaction packets. .
- Requester:
  - FMT: Defines the number of DWORD in Header field.
  - TYPE: Defines message versus memory, I/O, and configuration, and read or write. It also contains the r Field : Defined for message requester transaction packets for Implied Routing.
  - TC (Traffic Class): Defines the group of TLPs to be associated with per Flow Control protocol
  - TD: Defines that a Data field is attached.
  - EP: Defines the Data field has errors.
  - ATTRIBUTE: Defines the application of transaction ordering and snooping to the TLP
  - LENGTH: The number of DWORDs in the Data field.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
R	FMT 1 0		4		TYPE		0	R	2		TC		0	R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0											
7 BUS NUMBER 0								4 DEVICE # 0						2 FUNC# 0			
7 TAG 0								7 MESSAGE CODE 0									

## Transaction Packets for Message Address Space

- Message TLPs have certain bit fields in the Header field common to the memory and I/O address space and some additional bits defined as follows for requester and completer transaction packets. .
- Requester: .. continued
  - ATTRIBUTE: Defines the application of transaction ordering and snooping to the TLP
  - BUS#, DEVICE#, and FUNCTION#: Defined for ID Routing information in completer and configuration requester transaction packets. Defined as Requester ID.
  - TAG#: Provides a unique number to associate a specific requester transaction with a completer transaction.
  - MESSAGE CODE: Defined for message requester transaction packets for selecting the type of message.
- Completer: :No completer transactions are defined for the message address space.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	L	R	2	TC 0		R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG 0								3 LAST BE 0				3 FIRST BE 0			
31 ADDRESS 16															
15 ADDRESS 02 R R															

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	L	R	2	TC 0		R	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4	DEVICE # 0			2 FUNC# 0			
STATUS 2 0		B C M	11 BYTE COUNT 00												
7 BUS NUMBER 0								4	DEVICE # 0			2 FUNC# 0			
7 TAG # 0								R	6 LOWER ADDRESS 0						

## Single Versus Multiple Byte Accesses

- All of the TLPs read or write only one 32 data bit DWORD with the exception of memory requester transactions. That is, except for memory requester transactions the TLPs contain a Data field when applicable of a single DWORD.
- The TLPs associated with memory address space reads or writes may be a single DWORD or multiple DWORDs.
- In order to read or write a single or multiple DWORDs the following elements must be included in the Header field for the requester transaction packet:
  - FIRST BE [3::0] and LAST BE [3::0]:
  - LENGTH [9::0]
  - ADDRESS [31::00] or ADDRESS [63::00]
  - TAG# [7::0]
- In order to read or write a single multiple DWORDs the following elements must be included in the Header field for the completer transaction packet:
  - LENGTH [9::0]
  - BYTE COUNT [11::0]
  - LOWER ADDRESS [6::0]
  - TAG# [7::0]
- The use of the TAG# is in conjunction with the Requester ID and Completer ID are discussed in previous slides. The other elements are defined in the following slides.

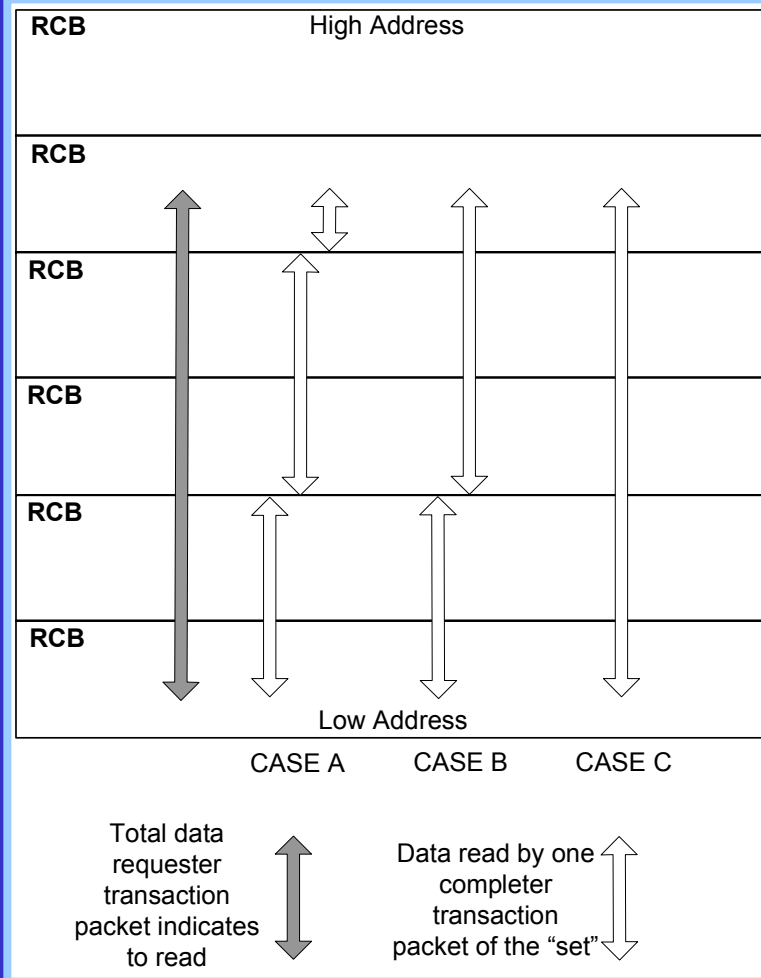


7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
R	FMT 1 0		4	TYPE		1	L	R	2	TC		0	R	R	R
T D	E P	ATTRI 1 0		R	R	9 LENGTH 0									
7 BUS NUMBER 0								4 DEVICE # 0				2 FUNC# 0			
7 TAG 0								3 LAST BE 0				3 FIRST BE 0			
31 ADDRESS 16															
15 ADDRESS 02 R R															

## Single Versus Multiple Byte Accesses ... continued

- Requester transaction packet:
  - FIRST BE [3::0]: Bytes enables of the first DWORD (lowest address order) in the Data field that selects which bytes are read or written.
  - LAST BE [3::0]: Bytes enables of the last DWORD (highest address order) in the Data field that selects which bytes are read or written. Not defined for a single DWORD Data field.
  - LENGTH [9::0]: For a memory read requester transaction packet it is the number of DWORDS to be read and returned in the associated completer transaction packet(s). For a memory write requester transaction packet it is the number of DWORDS in the Data field.
  - ADDRESS [31::00] or ADDRESS [63::00]: For a single DWORD read or write these bits in the requester transaction packet selects the DWORD in the memory address space. For multiple DWORDs read or write these bits in the requester transaction packet it selects the beginning address of multiple bytes to be accessed in memory address space.

## Single Versus Multiple Byte Accesses ... continued



- Completer transaction packet
  - The data associated with a single memory read requester transaction packet is read and sourced in either a “set” of one single completer transaction packet (CASE C), or a “set” of multiple completer transaction packets (CASE A & B).
  - The “set” of completer transaction packets is related to the Read Completion Boundary (RCB) which is a naturally aligned address boundary of 64 or 128 bytes. A memory resource may have multiple RCBs which provide address termination boundaries for multiple completer transaction packets of a “set”. Each completer transactions packets sourced must provide data to a RCB or to the remaining data if a single or last completer transaction packet.
  - For example, in CASE A the set of three completer transactions packets sourced with the data boundaries defined at the second and fourth RCB. In CASE B the set of two completer transaction packets are sourced with a data boundary at the second RCB.
  - As exemplified in CASE C all of the data is read and sourced in a single completer transaction packet containing multiple DWORDs. Because all of the data is read and source in a single packet, the RCBs are not a consideration.

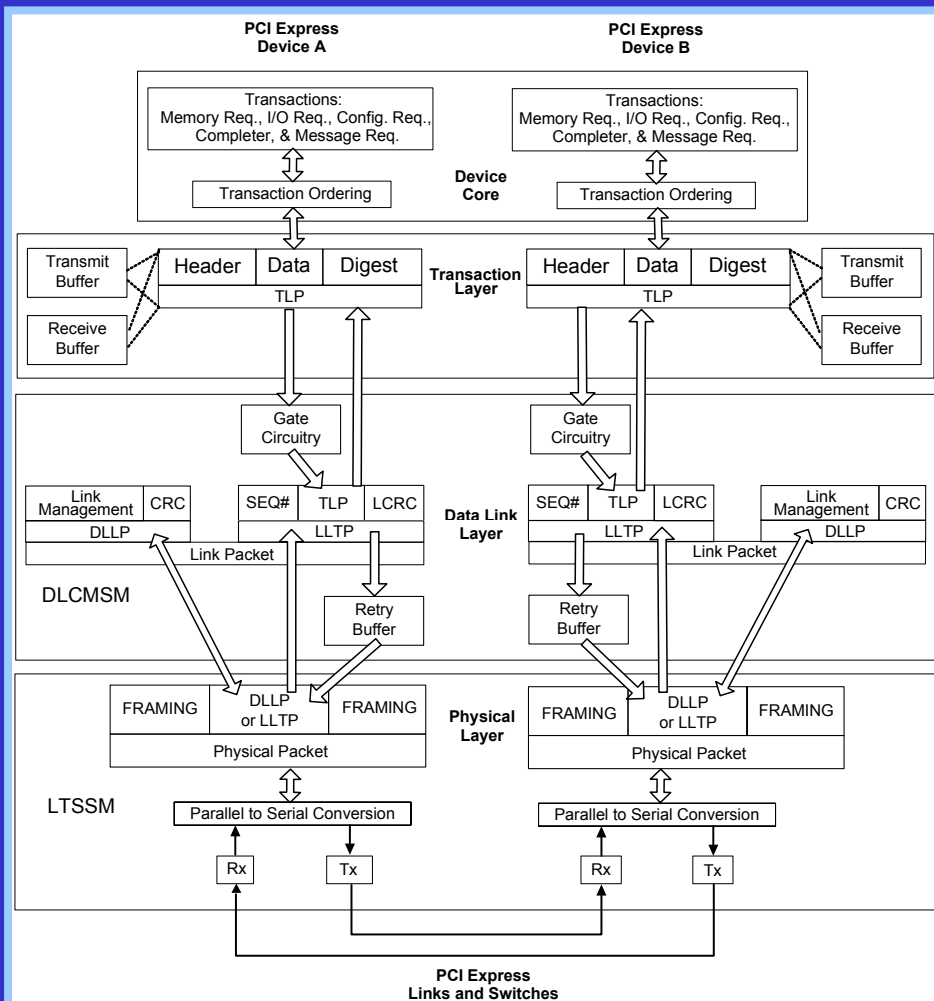
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0												
R	FMT 1 0		4		TYPE		1	L	R	2		TC	0	R	R	R	R										
T D	E P	ATTRI 1 0		R	R	9											LENGTH		0								
7								BUS NUMBER				0	4				DEVICE #		0	2		FUNC#	0				
STATUS 2 0		B C M		11											BYTE COUNT				00								
7								BUS NUMBER				0	4				DEVICE #		0	2		FUNC#	0				
7								TAG #				0	R	6											LOWER ADDRESS		0

## Single Versus Multiple Byte Accesses ... continued

- Completer transaction packet ... continued
  - LENGTH [9::0]: It is the number of DWORDs in the Data field.
  - BYTE COUNT [11::0] The total bytes that must still be read including those in the present completer transaction packet if part of a "set." If only one completer transaction packet is returned, the BYTE COUNT is the total valid bytes provided in the Data field. If multiple completer transaction packets are returned, the byte count represents the remaining bytes to be read in subsequent completer transaction packets plus the bytes in the present completer transaction packet.
  - LOWER ADDRESS [6::0] It defines the lower seven address bits of start address of the associated data. For the single completer transaction packet, the value of the two lowest order address bits.

# Chapter 7

## Data Link Layer & Packets



## Data Link Layer

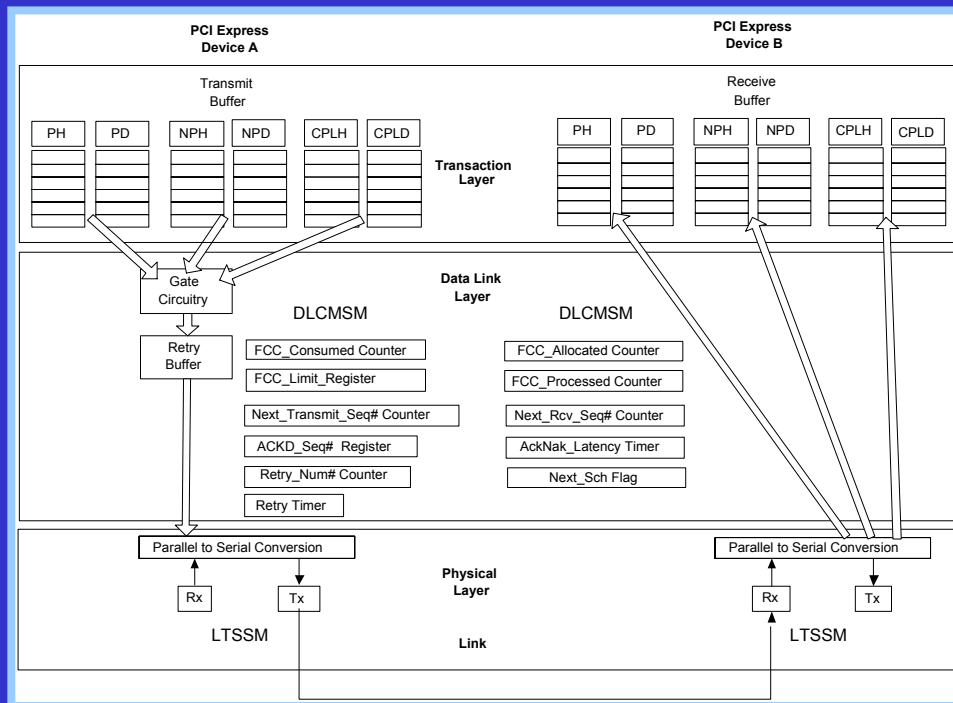
- As discussed in earlier slides there are three PCI Express layers in each PCI Express device that provides an interface between the PCI Express device core and the PCI Express link.
- The Data Link Layer provides the interface between the Transaction Layer and the Physical Layer that to transmit and receive TLPs transferred on the link.
- The two purposes of the Data Link Layers of each PCI Express device on a link is as follows:
  - To determine if buffer space is available to transfer a TLP from the Transaction Layer and subsequently across the link.
  - To ensure the integrity of the TLPs transferred across the link.
- In order to accomplish these two purposes the Data Link Layer implements the following:
  - Link activity states
  - Flow Control Initialization.
  - Flow Control Protocol
- In addition the Data Link Layer implements two types of packets: Data Link Layer Packets (DLLPs) associated with link management and Link Layer Transaction Packets (LLTPs) encapsulating TLPs.

### Data Link Layer ... continued

- The Link activity states are a sequence of logic states executed by the Data Link Control and Management State Machine (DLCMSM) within the Data Link Layer .
- The link activity states consists of three states: DL\_Inactive, DL\_Init (Initialization), and DL Active which define the Flow Control Initialization.
  - DL\_Inactive: The DLCMSM remains in this link activity state and returns to it whenever the link is not operational. The link is not operational if it can not transmit a DLLP and LLTP encapsulated within Physical Packets. The link's non-operational level is indicated by Physical LinkUp = "0" from the Link Training and Status State Machine (LTSSM) in the Physical Link Layer.
  - DL\_Init: The transition from the DL\_Inactive to the DL\_Init link activity state occurs when the link becomes operational as indicated by the Physical LinkUp = "1". During DL\_Init the DLCMSM in each PCI Express device on the link is trying to establish the initial available buffer space for VC0
  - DL\_Active: The transition from the DL\_Init link activity state occurs when the initial available buffer space at the receiving port for the LLTP is determined and reported to the transmitting port of the LLTP for Virtual Channel 0 (VC0). The Flow Control Initialization process continues for the remaining VCs.
  - The DLCMSMs of the Data Link layer exchanges link management information in the InitiFC1 and InitiFC2 DLLPs to determine buffer space initially available at the receiving port.

### Data Link Layer ... continued

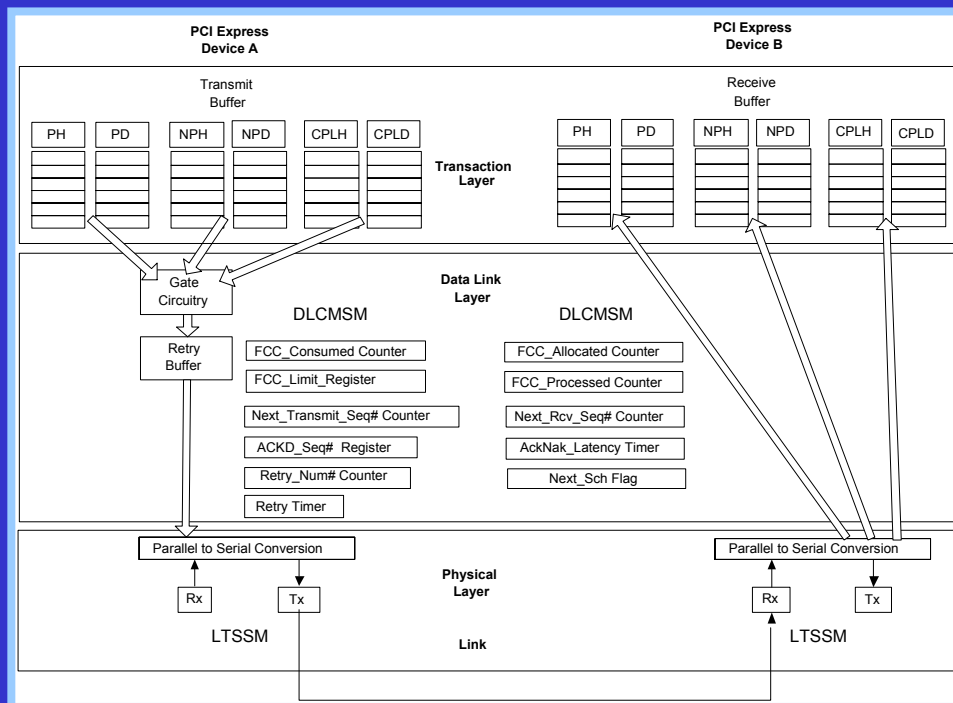
- When the **Flow Control Initialization** is successfully completed for VC0 the ability to transfer TLPs associated with VC0 from the Transaction Layer to the Data Link Layer is indicated. The indicator is the Link\_UP from the Data Link Layer to the Transaction Link Layer. Prior to the successful completion of the Flow Control Initialization of VC0, the indicator is Link\_DOWN.
- Subsequently, the **Flow Control protocol** updates the available buffer space and will not permit the Gate circuitry to transfer TLPs from the Transaction Layer unless buffer space is available at the TLPs' receiving port.
- The DLCMSMs in the TLPs' transmitting ports track the on-going available buffer space. The DLCMSMs in the receiving ports track the on-going buffer space consumed. The DLCMSMs exchange Update FC DLLPs as part of the **Flow Control protocol** to accomplish this tracking.
- TLPs can only be transferred from the Transaction Layer to the Data Link Layer if the associated VC number has completed Flow Control Initialization and buffer space is available at the receiving port. Each TLP is associated to a specific VC number via the Traffic Class (TC) number in the TLP's Header field. The mapping of the TC numbers to VC numbers occurs at the transmitting port.



## Data Link Layer ... continued

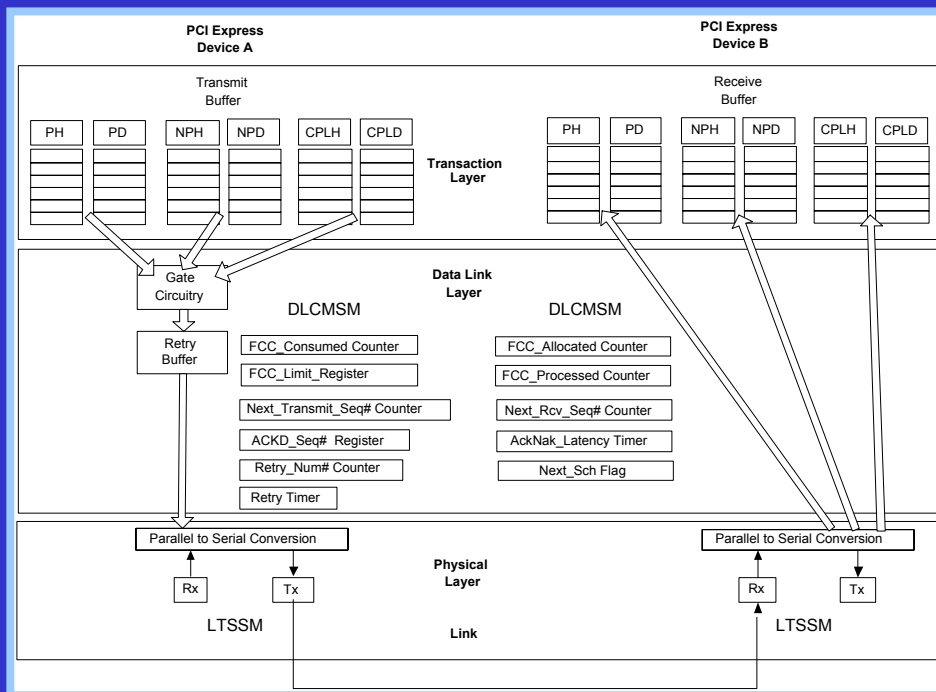
- The ongoing transfer of TLPs from Transaction Layer to the Data Link Layer within a PCI Express device for transmission is dependent on the available buffer space at the PCI Express device receiving them. The actual packet transmitted across the link is a Physical Packet containing a LLTP. The LLTP in turn contains the TLP. However, it is only the TLP that is buffered at the receiving port.
- The available buffer space at the receiving port (exemplified by Device B) of the LLTP and thus the TLP is defined for each VC number, each type of TLP, and Header field versus the Data field.
- For a TLP assigned to a specific VC number. to transfer from the Transaction Layer to the Data Link layer at the transmitting port (exemplified by Device A) requires that sufficient buffer space is available at the receiving port of the LLTP that contains the TLP.
- In that the buffer space is defined per VC number, it is possible that only TLPs of a specific VC number or VC numbers can be transferred at any specific time. However, transaction ordering, can never be violated.





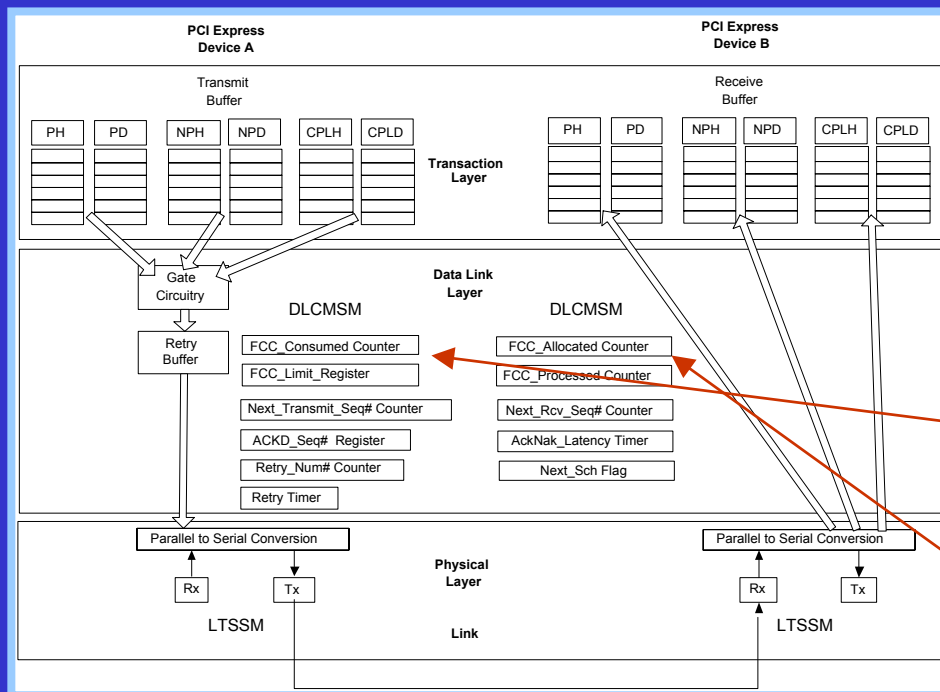
## Data Link Layer ... continued

- Each VC is assigned a specific set of buffers. The set of buffers at each receiving port for each VC number is defined as follows:
  - Buffer type PH (posted header):** Buffer space for Header fields of memory write and message requester transaction packets.
  - Buffer type PD (posted data):** Buffer space for Data fields of memory write and message requester transaction packets.
  - Buffer type NPH (non-posted header):** Buffer space for Header fields of memory read, I/O read and write, and configuration read and write requester transaction packets.
  - Buffer type NPD (non-posted header):** Buffer space for Data fields of I/O write and configuration write requester transaction packets.
  - Buffer type CPLH (completer header):** Buffer space for Header fields of completer transaction packets.
  - Buffer type CPLD (completer data):** Buffer space for Data fields of completer transaction packets.



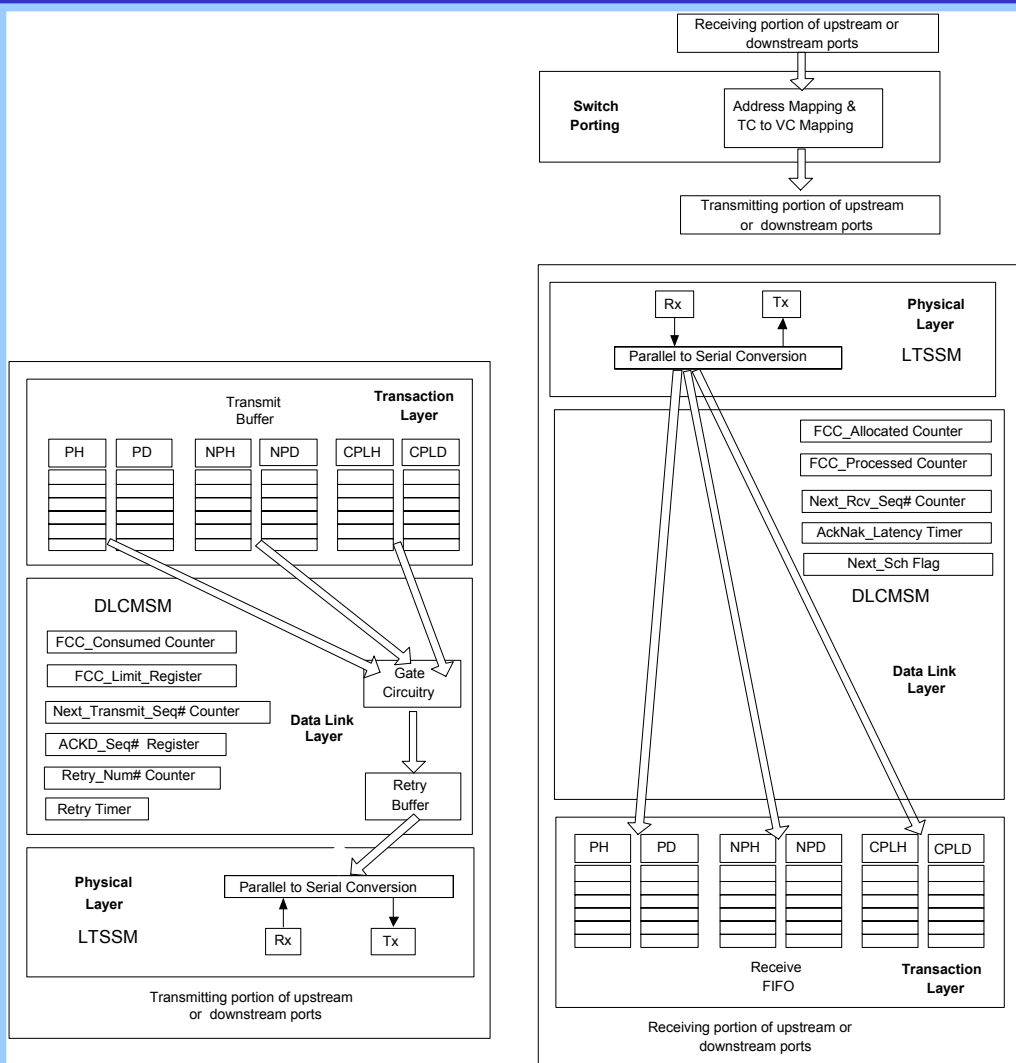
## Data Link Layer ... continued

- For a TLP assigned to a specific VC number to be transferred from the Transaction Layer to the Data Link Layer, the following must occur:
  - The buffer space must be available for both the Header field and the Data field for the specific type of TLP.
  - The determination of available buffer space and buffer space required for a TLP is measured by Flow Control Credits (FCC).
  - Once the TLP is transferred it is encapsulated by the Data Link Layer into a LLTP, and the LLTP is transferred to the Physical Layer.



## Data Link Layer ... continued

- FCC unit of measure is 16 bytes. The Header field of a TLP for memory, I/O, configuration, or message requester transactions; and associated completer transactions requires one FCC. The FCCs needed by the Data field is the actual number of bytes of data to be transferred divided by 16.
- The DLCMSMs in the transmitting ports (exemplified by Device A) track the available buffer space. The associated quantities are defined in the FCC\_Consumed Counter and FCC\_Limit Register.
- The DLCMSMs in the receiving ports (exemplified by Device B) track the available buffer space. The associated quantities are defined in the FCC\_Allocated Counter and FCC\_Processed Register.



## Data Link Layer ... continued

- The previous slides have focused on the Data Link Layer at the source and destination of the requester and completer transaction packets.
- The switches must port the TLPs associated with these requester and completer transaction packets from the receiving to the transmitting ports.
- In each switch the TLPs must be extracted from the LLTPs contained within the Physical Packets.
- The transmission of these Physical Packets requires sufficient buffer space at the receiving port of the switch. The protocol to determine that sufficient buffer space is available at the receiving port is the same as for the requester or completer destination.
- The received TLPs are ported to the correct transmitting port by the Address Mapping & TC to VC Mapping.
- At the transmitting port the protocol to determine if the TLPs can be encapsulated into LLTPs to be transmitted in Physical Packets is the same as transmitting port of a requester or completer source.

## Packets of the Data Link Layer

- As previously stated there are two types of packets implemented by the Data Link Layer : Data Link Layer Packets (DLLPs) and Link Layer Transaction Packets (LLTPs)
- The DLLPs only define a Header field and are used for link management by the DLCMSM of the Data Link Layer relative to the following activities.
  - Flow Control: InitFC1, InitFC2, and UpdateFC DLLPs.
  - Acknowledgement: Ack and Nak DLLPs
  - Power Management: PM\_Active\_State\_Request\_L1, PM\_Enter\_L1, PM\_Enter\_L23, and PM\_Request\_Ack DLLPs.
  - Vendor-Specific: Vendor Specific DLLPs.
- Flow Control DLLPs: As discussed in previous slides these DLLPs are exchanged to determine the size of available buffer space at the LLTPs' receiving ports. Further discussion in later slides. The format is as follows:
  - LINK TYPE [7::0]: Defines type of DLLP
  - HdrFC [7::0]: Buffer size information relative to the Header field of TLPs.
  - DataFC [11::0]: Buffer size information relative to the Data field of TLPs.
  - CRC [15::00]: Cyclic redundancy checking information over the DLLP.

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
7 LINK TYPE 0								R	R	7 HdrFC 2					
HdrFC 1 0		R	R	11 DataFC 0											
15 CRC 00															

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
7 LINK TYPE 0								R	R	R	R	R	R	R	R
R	R	R	R	11 AckNakSEQ# 0											
15 CRC								00							

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Link Type							0	R	R	R	R	R	R	R	R
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
CRC PAW															00

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
7                      Link Type                      0								7                      Vendor-Specific                      0							
15															

## Packets of the Data Link Layer ... continued

- Acknowledgement DLLPs:** The purpose of these DLLPs is for the receiving port to acknowledge successful (Ack) or unsuccessful (Nak) receipt of a LLTP. Further discussion in later slides. The format is as follows:
  - LINK TYPE [7::0]: Defines the type of DLLP
  - AckNakSEQ# [11::0]: SEQ# of associated LLTP
  - CRC [15::00]: Cyclic redundancy checking information over the DLLP
- Power Management DLLPs:** The purpose of these DLLPs is for the link and associated PCI Express devices to transition to the L1 (lower power) L2/3 Ready (sleep or powered-off) Further discussion in later slides. The format is as follows:
  - LINK TYPE [7::0]: Defines the type of DLLP
  - CRC [15::00]: Cyclic redundancy checking information over the DLLP
- Vendor-defined DLLPs** The purpose of this DLLP is to provide vendors implementation specific DLLPs :
  - LINK TYPE [7::0]: Defines the type of DLLP
  - Vendor-Specific [7::0]: RESERVED and specific to vender
  - VENDOR-SPECIFIC [15::0]: RESERVED and specific to vender
  - CRC [15::00]: Cyclic redundancy checking information over the DLLP

7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0		
15								SEQ#								00	
15								TLP								00	
15								TLP								00	
<div><div>V</div><div>V</div><div>Multiple DWORDS associated with one TLP</div><div>V</div><div>V</div></div>																	
15								TLP								00	
31								LCRC								16	
15								LCRC								00	

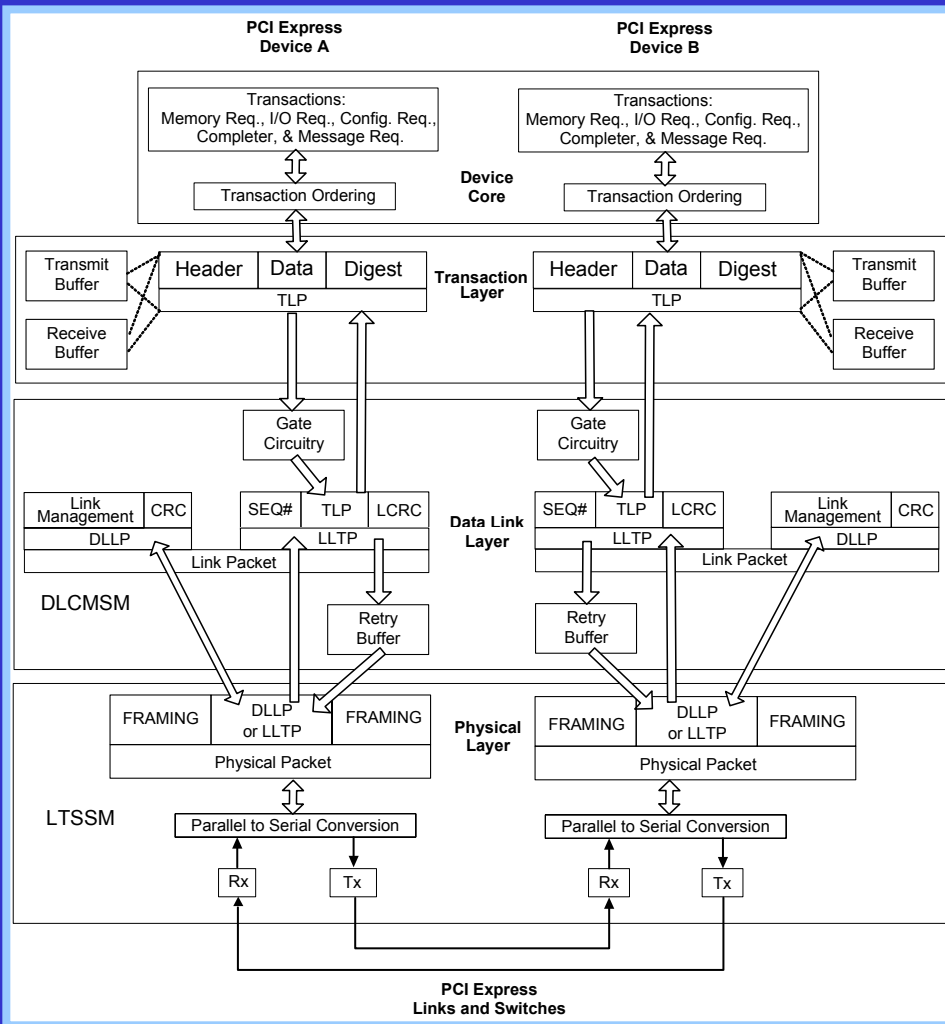
### Packets of the Data Link Layer ... continued

- As previously stated there are two types of packets implemented by the Data Link Layer : Data Link Layer Packets (DLLP) and Link Layer Transaction Packets (LLTPs).
- The TLP is encapsulated into a LLTP by the Data Link Layer with the addition of
  - SEQ# [15::00]: Provide a specific sequential number to maintain strong ordering of LLTPs across the link.
  - TLP is the intact packet from the Transaction Layer Packet.
  - LCRC [31::00]: Cyclic redundancy checking information over the LLTP.

# Chapter 8

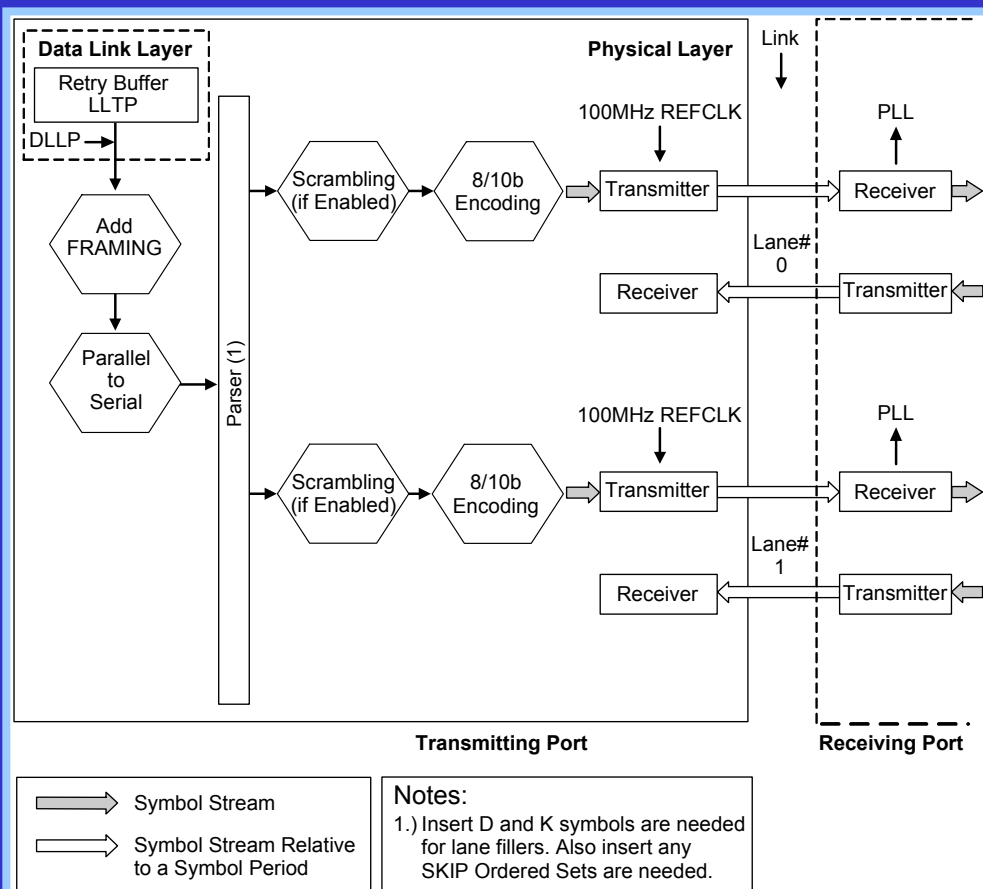
## Physical Layer & Packets





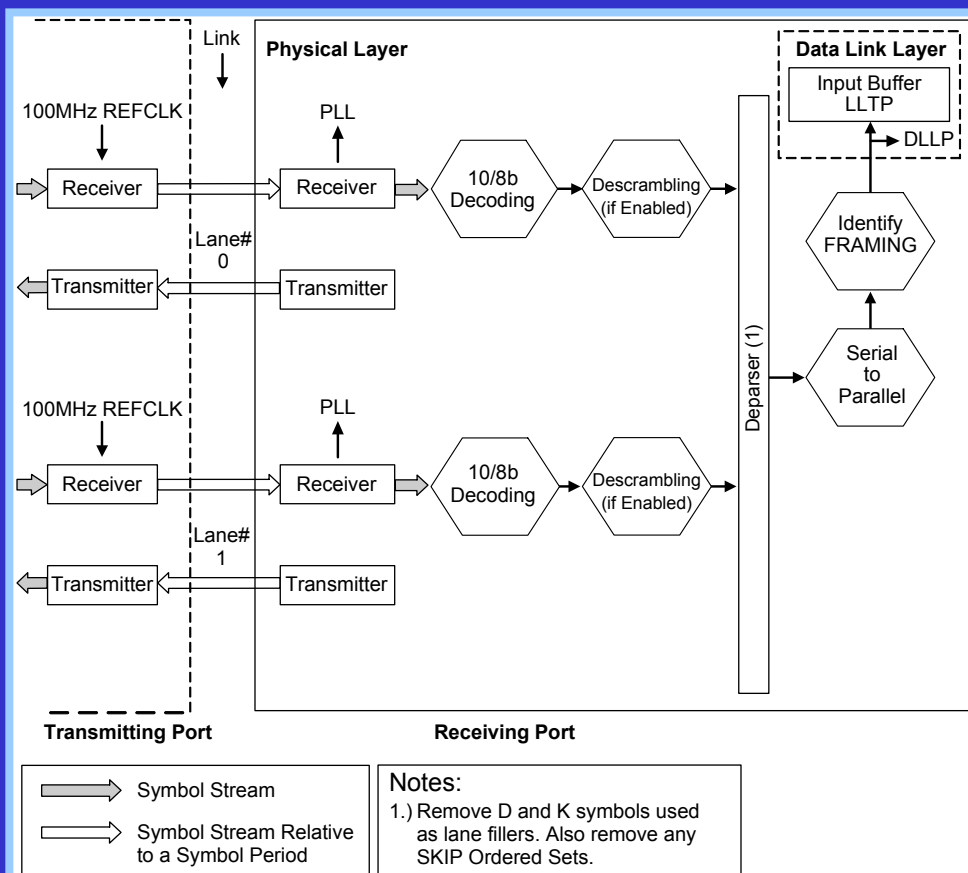
## Physical Layer

- As discussed in earlier slides there are three PCI Express layers in each PCI Express device that provides an interface between the PCI Express device core and the PCI Express link.
- The Physical Layer provides the interface between the Data Link Layer and the link to transmit and receive Physical Packets transferred across the link.
- The Physical Layer supports different link states with the Link Training and Status State Machine (LTSSM).



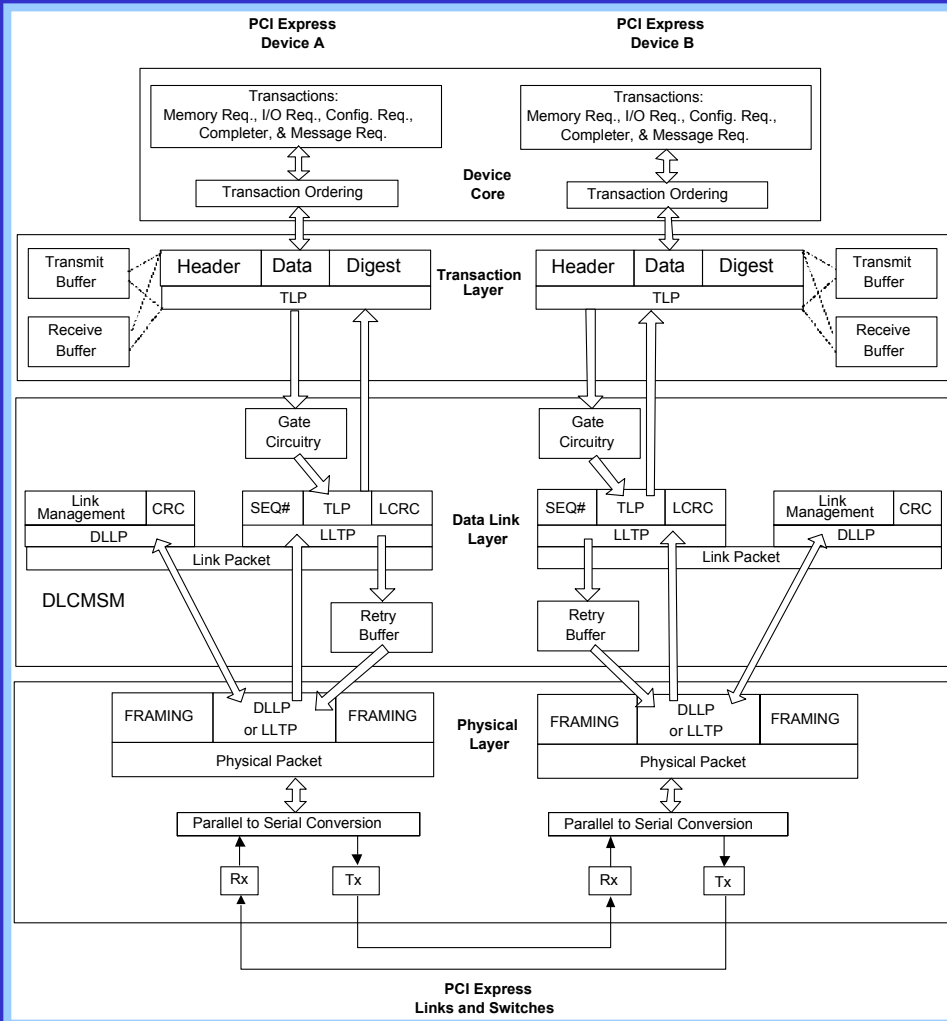
## Physical Layer ... continued

- The purpose of the Physical Layer of each PCI Express device's transmitting port of the TLPs on a link is as follows:
  - Convert the parallel orientation of the LLTP and DLLP to the serial orientation of the Physical Packet at the transmitting port ... and vice versa at receiving port.
  - Encoding the series of bytes of the Physical Packet into a stream of symbols with an integrated reference clock at the transmitting port.
  - Parse the stream of symbols across the multiple lanes in a link.
  - Execute link configuration via Link Training.



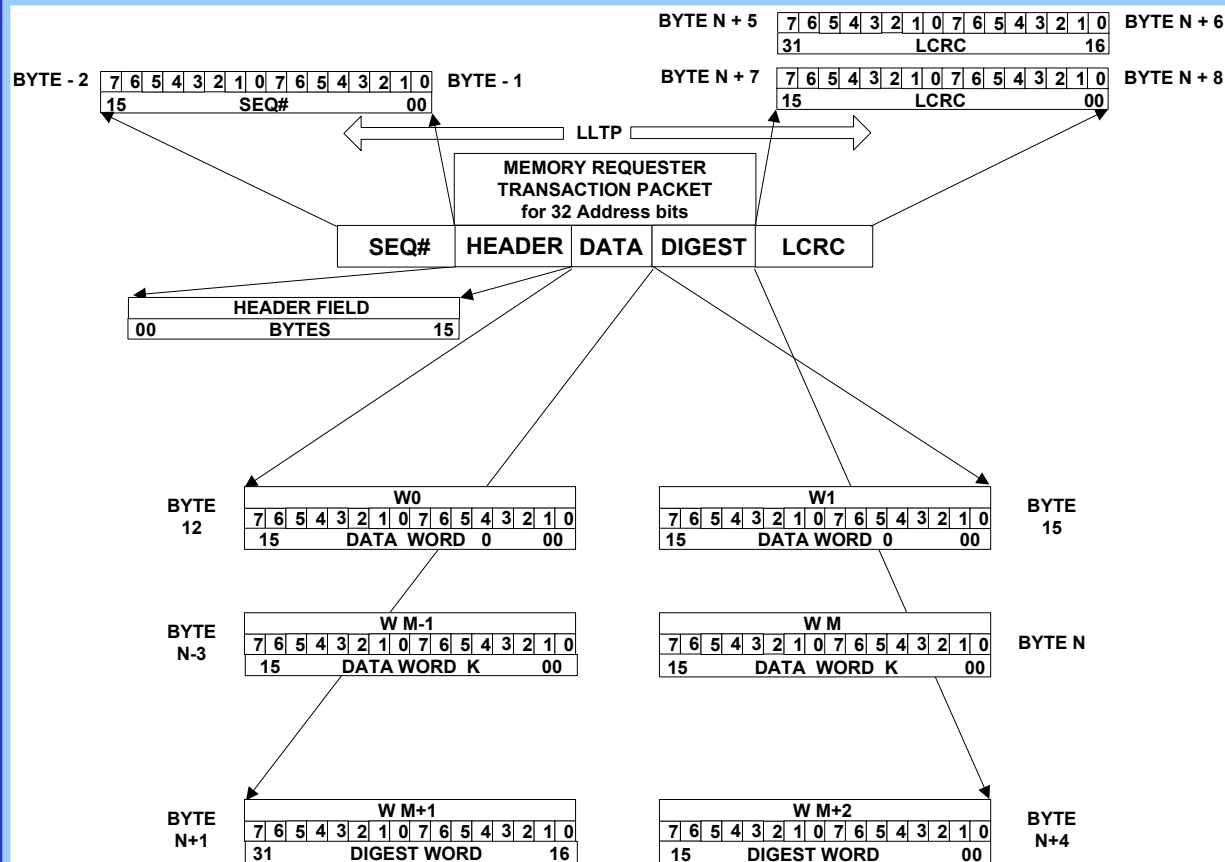
## Physical Layer ... continued

- The purpose of the Physical Layer of each PCI Express device's receiving port of the TLPs on a link is as follows :
  - Convert the serial orientation of the Physical Packet to the parallel orientation of the LLTP and DLLP at the receiving port.
  - Decode the stream of symbols with the extraction of the reference clock into a series of bytes of the Physical Packet .
  - Deparse of the stream of symbols from multiple lanes in a link.
  - Execute link configuration via Link Training.



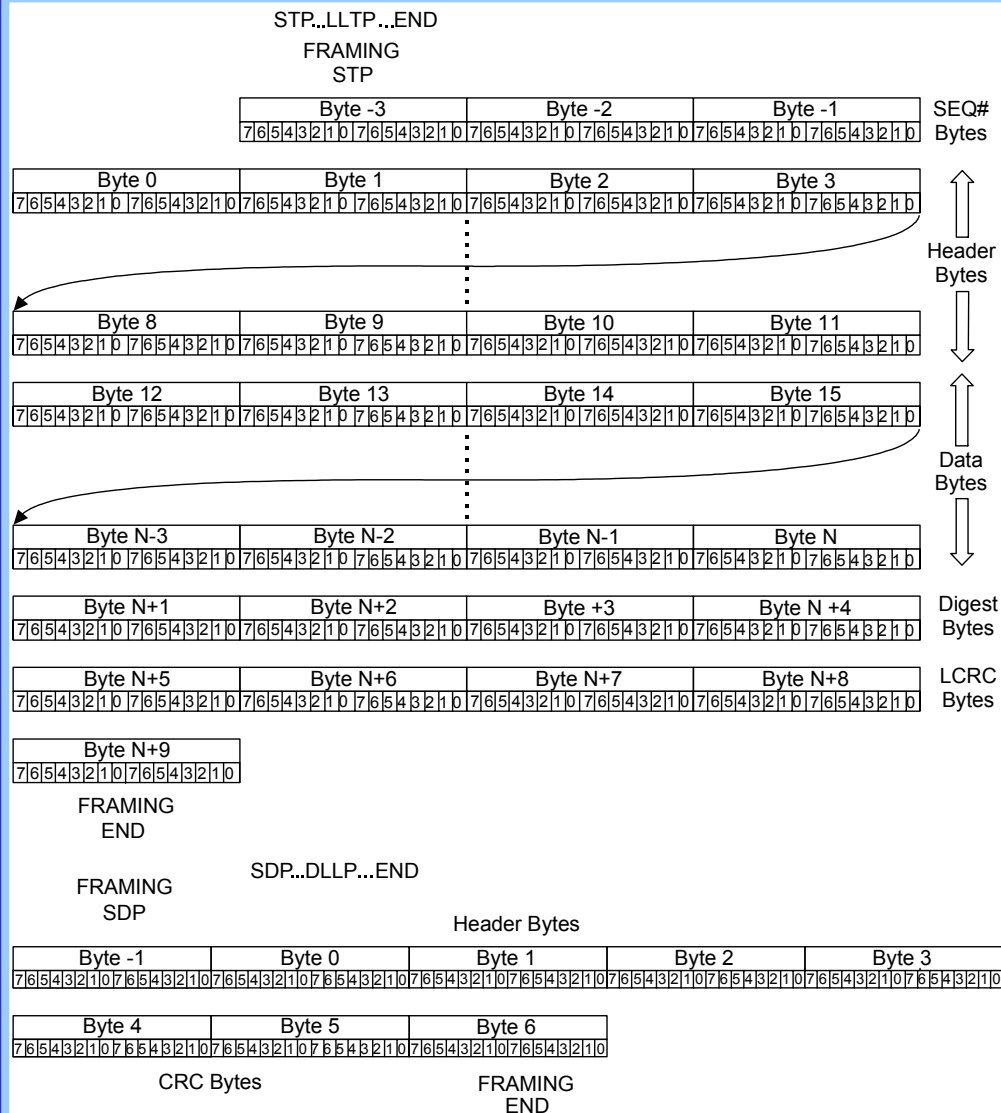
## Physical Layer ... continued

- In order to accomplish the purposes outlined in the previous slides the Data Link Layer implements the following:
  - The Physical Layer defines two types of Physical Packets. One type encapsulating LLTPs and the other encapsulating the DLLPs
  - 8/1b encoding and PLL
  - Ordered Sets
  - Link Training



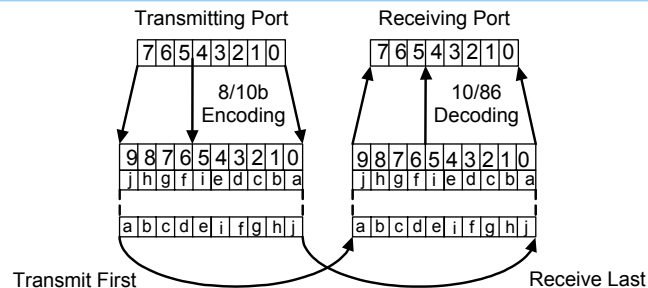
## Physical Packets

- Using a LLTP encapsulated in a Physical Packet as an example:
  - The parallel orientation of the LLTP is defined in terms of WORDs and DWORDs for the Data field .
  - The serial orientation of the Physical Packet is defined as a series of bytes. The series of bytes can also be viewed as a series of bits.
  - Difference in the series of bytes for the LLTP versus the Physical Packet is addition of FRAMING bytes in the Physical Packet.

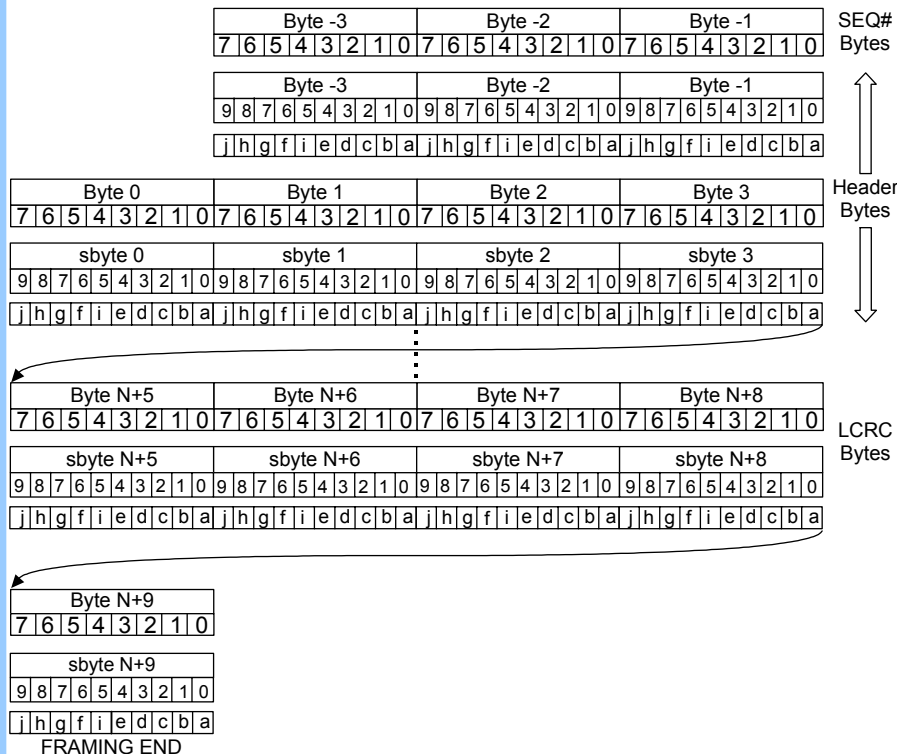


## Physical Packets

- The FRAMING bytes provides the following information:
  - The FRAMING bytes at the beginning and end of a series of bytes used for the LLTP identify and define the boundaries of a specific LLTP.
  - A LLTP may contain a nullified TLP. A different set of FRAMING bytes are used at the end to identify the the LLTP that contains a nullified TLP.
  - FRAMING bytes at the beginning and end of a series of bytes are also used to define the boundaries of a specific DLLP.

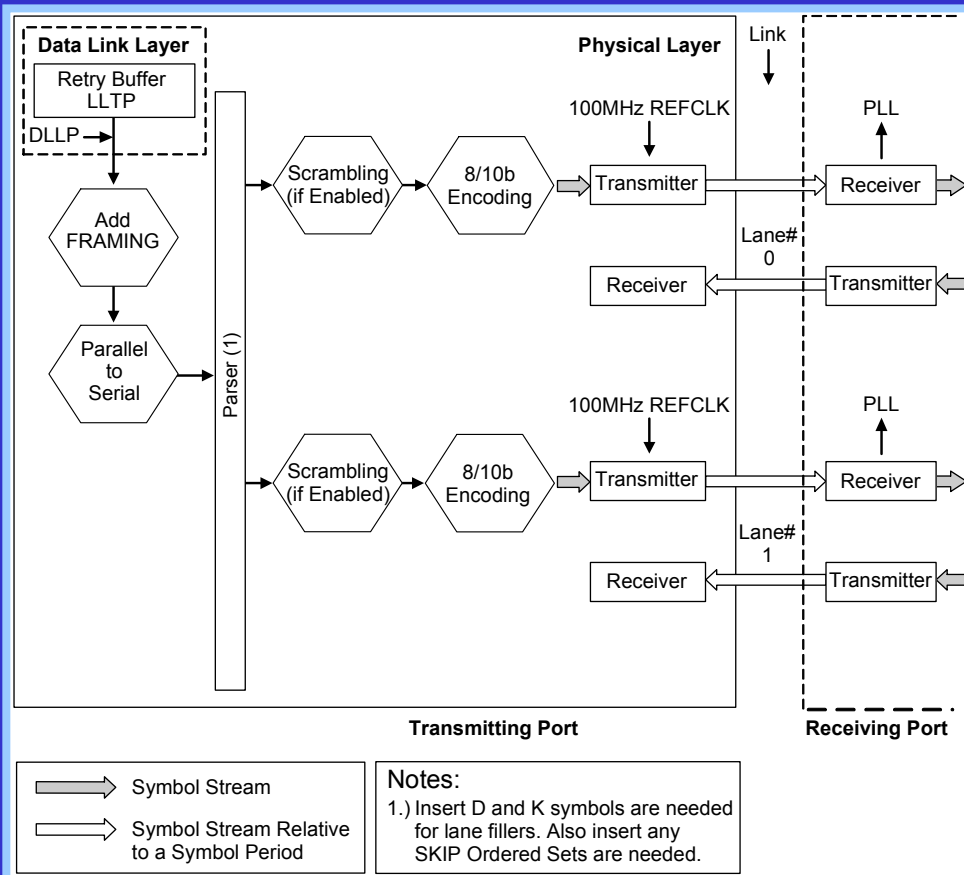


#### FRAMING STP



## Encoding and Decoding

- The series of bytes can also be viewed as a series of bits. Each byte consist of 8 bits.
- In order to integrate at the transmitting port a reference clock the 8 bit bytes must be encoded into 10 bit sbytes. Each sbyte is called a symbol.
- The conversion from 8 bit bytes to 10 bit sbytes to integrate the reference clock is called 8/10b encoding.
- The resulting symbol stream consists of a stream of bits. The stream of bits is transmitted across the link beginning with sbyte-3 (per this example). Within each sbyte the bit labeled “a” is transmitted first.
- At the receiving port the PLL extracts the reference clock to determine the valid bit periods within the sbytes of the symbol stream
- The conversion from 10 bit sbytes to 8 bit bytes is called 10/8b decoding.



## Lane Parsing and Deparsing of Symbol Stream

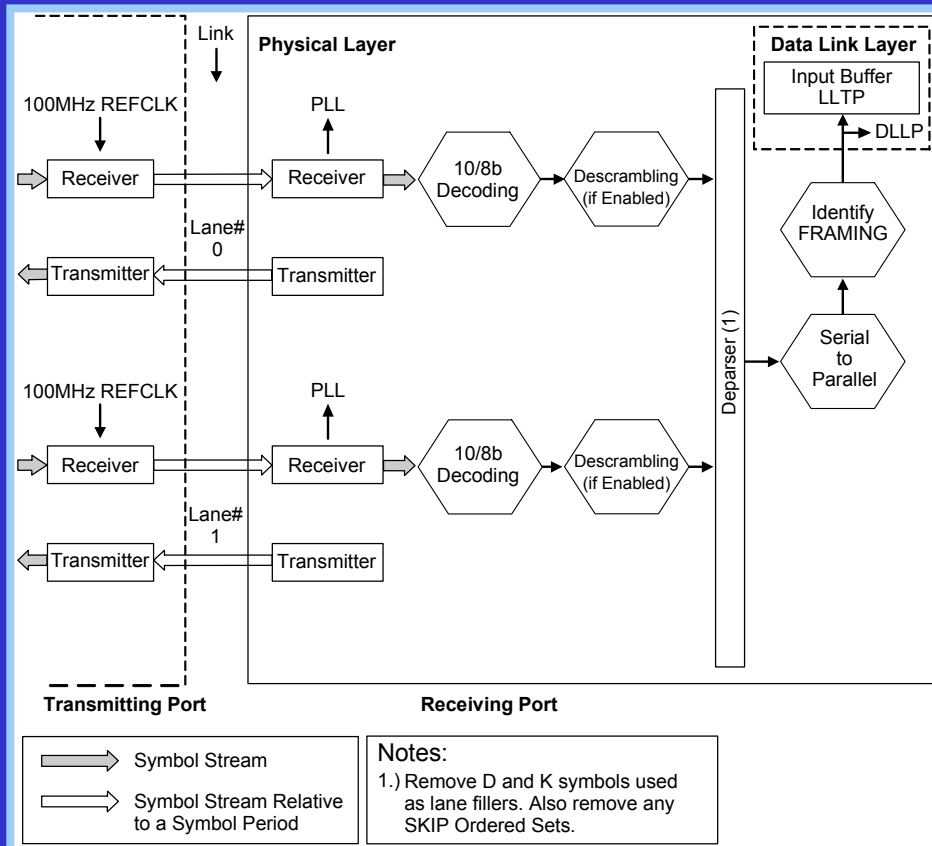
- In the simplest implementation with one lane on a link, the symbol stream is only transmitted across a link.
- When multiple links are configured in a link, the symbol stream is parsed across all configured lanes aligned in symbol periods.



LANE #	beginning	Symbol Periods				continue
	0	1	X	X+1	X +2	X+ 3
0	I D symbol	S - 3 STP <sub>K</sub> symbol	S N LCRC D symbol	S 3 DLLP D symbol	S -3 STP <sub>K</sub> symbol	S 5 LLTP D symbol
1	I D symbol	S - 2 SEQ# D symbol	S N+1 LCRC D symbol	S 4 CRC D symbol	S - 2 SEQ# D symbol	S 6 LLTP D symbol
2	I D symbol	S - 1 SEQ# D symbol	S N+2 LCRC D symbol	S 5 CRC D symbol	S - 1 SEQ# D symbol	S 7 LLTP D symbol
3	I D symbol	S 0 LLTP D symbol	S N+3 END K symbol	S 6 END K symbol	S 0 LLTP D symbol	S 8 LLTP D symbol
4	I D symbol	S 1 LLTP D symbol	S -1 SDP K symbol	PAD K symbol	S 1 LLTP D symbol	S 9 LLTP D symbol
5	I D symbol	S 2 LLTP D symbol	S 0 DLLP D symbol	PAD K symbol	S 2 LLTP D symbol	S 10 LLTP D symbol
6	I D symbol	S 3 LLTP D symbol	S 1 DLLP D symbol	PAD K symbol	S 3 LLTP D symbol	S 11 LLTP D symbol
7	I D symbol	S 4 LLTP D symbol	S 2 DLLP D symbol	PAD K symbol	S 4 LLTP D symbol	S 12 LLTP D symbol

### Lane Parsing and Deparsing of Symbol Stream .. continued

- The Physical Packets are transmitted back to back across the multiple lanes when possible,. However, when this is not possible, the configure lanes are filled with the IDLE (I) D symbol. The IDLE D symbol provides the reference clock ensure the PLL at the receiving port retain bit and symbol lock.
- It is possible that not all Physical Packets will perfectly fill all configured lanes across all symbol periods. The PAD K symbol is used as a lane filler when the Physical Packet will not fill all lanes of the last symbol period of the parsed Physical Packet.



## Lane Parsing and Deparsing of Symbol Stream ... continued

- At receiving port the reverse procedure occurs
- Any lane fillers are removed
- The symbol stream is deparsed from all configured lanes to a single symbol stream.

## Ordered Sets

- In addition to the transmission of Physical Packets and lane fillers, the transmitters will transmit Ordered Sets (OSs) across the lanes of the link. The Skip, FTS, and Electrical Idle OSs consist of 4 symbols. TS 1 and 2 OS consists of 16 symbols.
  - Skip OS: The data bit rate frequency and timing tolerance among all symbols transmitted on parallel configured lanes is 0 ppm. However, the data bit rate frequency for transmitting end of the link has a nonzero tolerance relative to the frequency bit rate at the receiving end of the link. This OS is transmitted to to compensate for the tolerance.
  - FTS (Fast Training Sequence) OS: This OS is required to achieve bit and symbol lock in the transition from L0s to L0 link states. In the L0s link state the bit and symbol lock are lost because no symbols are transmitted on the link for a specific direction to lower power.
  - Electrical Idle OS: The transition between certain link states requires the transmitters to enter to electrical idle (high impedance). In most cases the transition to electrical idle first requires the transmitters to transmit the Electrical Idle OS to indicate the pending electrical idle to the other port on the link with exceptions.
  - TS (Training Sequence) OS: There are two sets: TS1 and TS2. The primary purpose of the TS OSs provide information used in Link Training. This information includes link and lane numbers, the data bit rate to be used for transmission. For other purposes other information provided controls data scrambling, and transitions to the Hot Reset, Disable, and Loopback link states.

## Link Training

- The link between two PCI Express devices are a set of differentially driven signal lines. The number of signal lines establishes the number of lanes and is not a fix value. Each pair of PCI Express devices on a specific link can have unique number of lanes relative to other links in the PCI Express platform.
- Before Physical Packets can be transmitted across a link the exact number of lanes must be established. Such an establishment provides a set of configured lanes that define a configured link.
- Link Training is defined as sequencing through the Detect, Polling, and Configuration link states in order to configure a set of lanes to define a configured link. The three link state defined for the Link Training Status State Machine in the Physical Layer is as follows:
  - **Detect link states:** Establish the existence of a PCI Express device on each end of link. That is, detect the lanes within a link common to both PCI Express devices on the link.
  - **Polling link state:** Establish the bit and symbol lock, lane polarity inversion, and highest common data bit rate on the detected but yet-to-be- configured lanes that exist between the two PCI Express devices.
  - **Configuration link state:** Some or all of the detected lanes that successfully complete the Polling link states are processed into configured lanes. Those lanes that can be detected but cannot successfully complete the Polling link state cannot be processed onto a configured lane. Configured lanes are collected into configured links in the link sub-states of the Configuration link state. These link sub-states consequently establish the link width and lane ordering with support of lane reversal. Finally, these link sub-states also implement lane-to-lane de-skew and the N\_FTS value is also established.

# Chapter 9

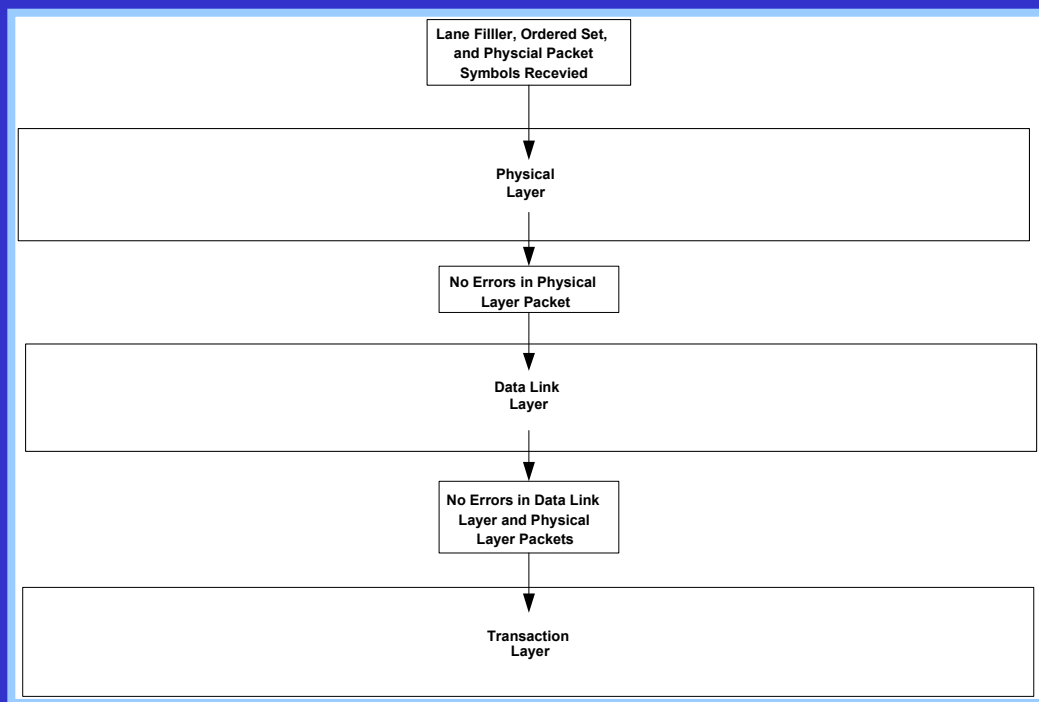
## Errors

## Errors

- The execution of transactions between PCI Express devices cores may be adversely affected on several levels resulting in errors.
- PCI Express defines three types of errors
  - **Correctable Errors:** Occurs without any loss of information. Responded to by hardware and not by the Error software
  - **Nonfatal Errors:** Not correctable but is isolated to a specific link and the two PC Express devices on the link. Responded to by the Error software and not by the hardware.
  - **Fatal Errors:** Fatal errors are not correctable and may or ay not be isolated to a specific links or PCI Express devices. Responded to by the Error software and not by the hardware.
- The source of the errors are related to the three PCI Express layers as follows:
  - In the Transaction Layer: Fatal and nonfatal errors can occur reflecting the following:
    - Malformed packet (incorrect TLP format), Unsupported Requester (TLP received not supported by PCI Express device), or Unexpected Completer (TLP received that was not expected).
    - Receiver Overflow reflects that for whatever reason the receiving buffer of the TLPs are overflowing.
    - Completer Timeout reflects that a completer transaction packet was not received in a reasonable time relative to the associated requester transaction packet being transmitted.
    - Completer Abort reflects that the destination of the requester transaction packet can not properly process the TLP that was received without other errors.
    - Flow Control Cyclic Protocol Error reflects a error in the protocols of Flow Control Initialization or ongoing Flow Control protocol that addresses the available buffer space at the TLPs' receiving port.

## Errors ... continued

- The source of the errors are related to the three PCI Express layers as follows: ... continued
  - In the Data Link Layer: Correctable, fatal, and nonfatal errors can occur in the layer reflecting the following:
    - BAD LLTP or BAD DLLP reflects a cyclic redundancy error or a SEQ# problem.
    - Retry\_Timer Timeout and Retry\_Num# reflects the effort by transmitting port to transmit the contents of the retry buffer and receive acknowledgement from the LLTPs' receiving port within a limited time. If the LLTPs' transmitting port has too many Retry\_Timer Timeouts reflecting non-receipt of an acknowledgement and the Retry\_Num# Rollover error occurs.
    - Data Link Layer Protocol Error reflects a error in the protocols of Flow Control Initialization or when the value of the AckNakSEQ# in the DLLPs does not equal an unacknowledged LLTP in the retry buffer.
  - Physical Layer: Correctable, fatal, and nonfatal errors can occur in the layer reflecting the following:
    - Receiving Error reflects that a symbol is received that is not valid or has a disparity error.
    - Training Error reflects a error in Link Training protocol Execution of Detect, Polling and Configuration link states.
- In the following slides the focus is on the basic errors that are detected with the receipt of a symbol stream containing the Physical Packets, LLTPs, DLLPs, and TLPs. The Book provides more extensive details on all errors.



## Layers ... Processing Errors and Reporting Overview

- If there were no possibilities of errors the processing of the Physical Packets received would simply consist of deparsing the symbol stream and extraction of the reference clock. The Data Link Layer would simply have to extract the LLTPs and DLLPs from the Physical Packets. The Transaction Layer would directly port the TLPs' contents to the PCI Express device core. In the case of a switch the TLPs are ported to another port.
- BUT ... In the real world there are possible errors relative to each layer; consequently the processing of stream of symbols received includes processing errors in the Physical Packets, LLTPs, DLLPs and TLPs at each layer.



Lane Filler, Ordered Set,  
and Physical Packet  
Symbols Received

Physical  
Layer

No Errors in Physical  
Layer Packet

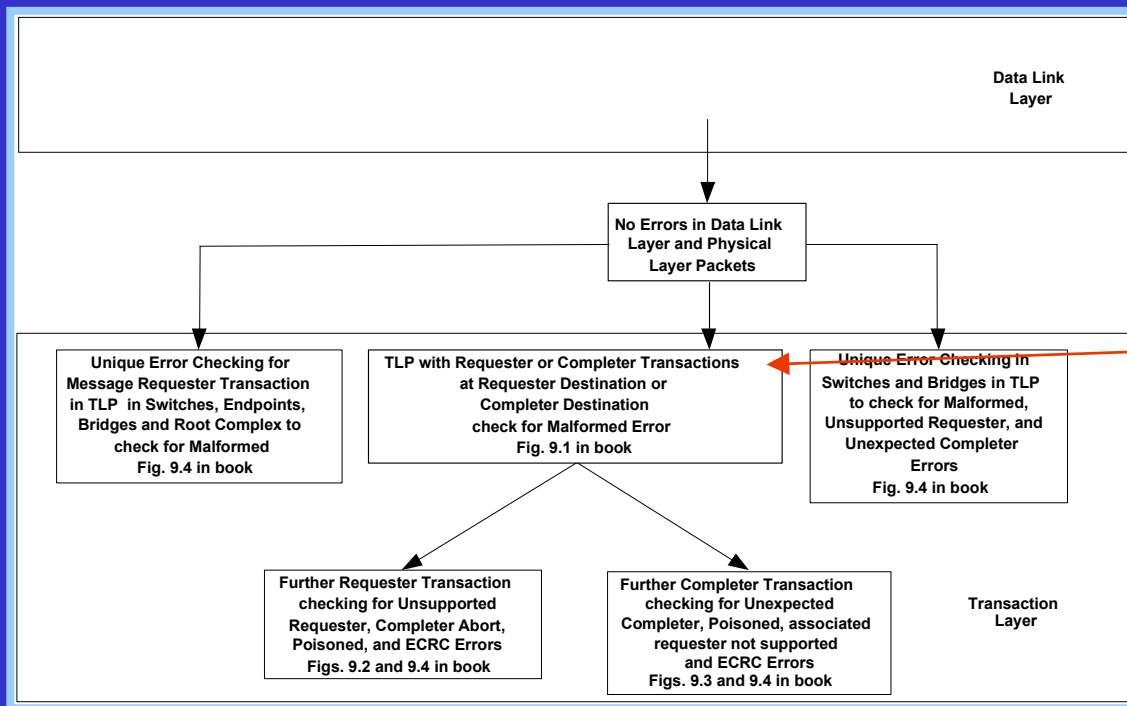
Data Link  
Layer

No Errors in Data Link  
Layer and Physical  
Layer Packets

Transaction  
Layer

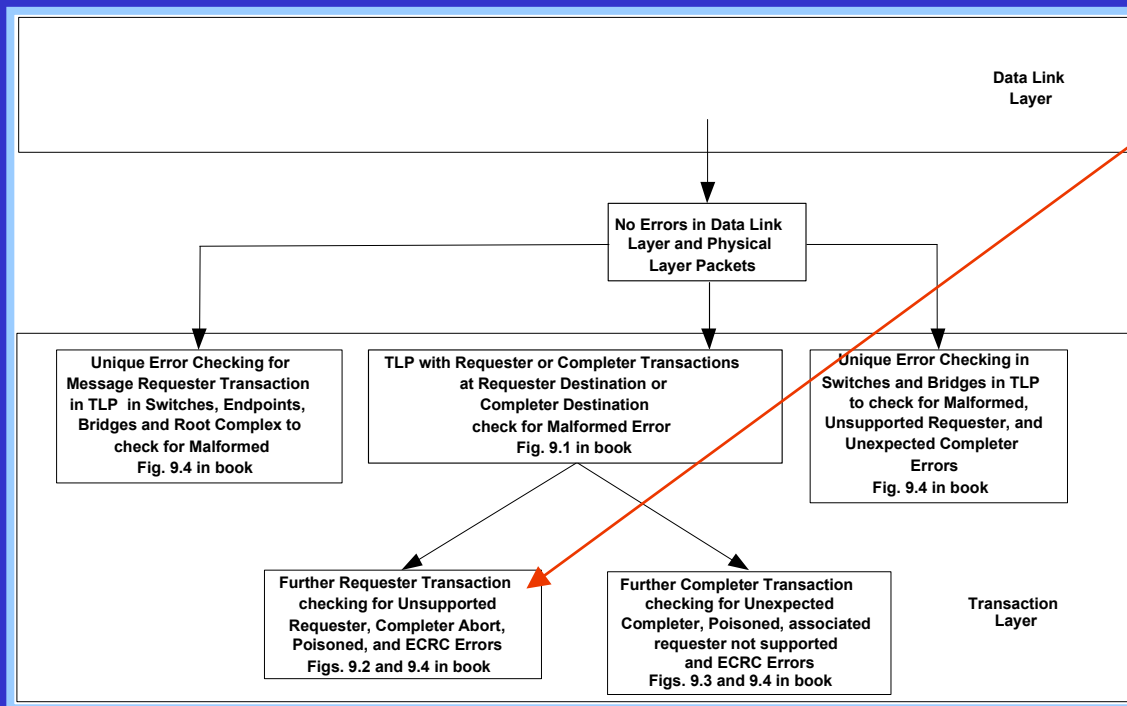
## Layers ... Processing Errors and Reporting Overview .. continued

- To pass the DLLPs and LLTPs from the Physical Layer to the Data Link Layer, the Physical Packets must be error free.
- To execute the link management in by the Data Link Layer the DLLPs must be error free.
- To pass TLPS from the Data Link Layer to the Transaction Layer, the TLP must be error free.
- To pass the transaction information from the Transaction Layer to the PCI Express core, the TLP must be error free.
- To simplify the discussion flow, the following slides will first focus on the error detection relative to the Transaction Layer which is fairly self-contained, followed by error detection relative to the Data Link Layer which includes confirming receipt of LLTPs, and then error detection in the Physical Layer.
- Each of the layers detect specific errors, but all are reported per a common protocol with either Minimal Or Advance Error reporting .. Discussed in later slides.



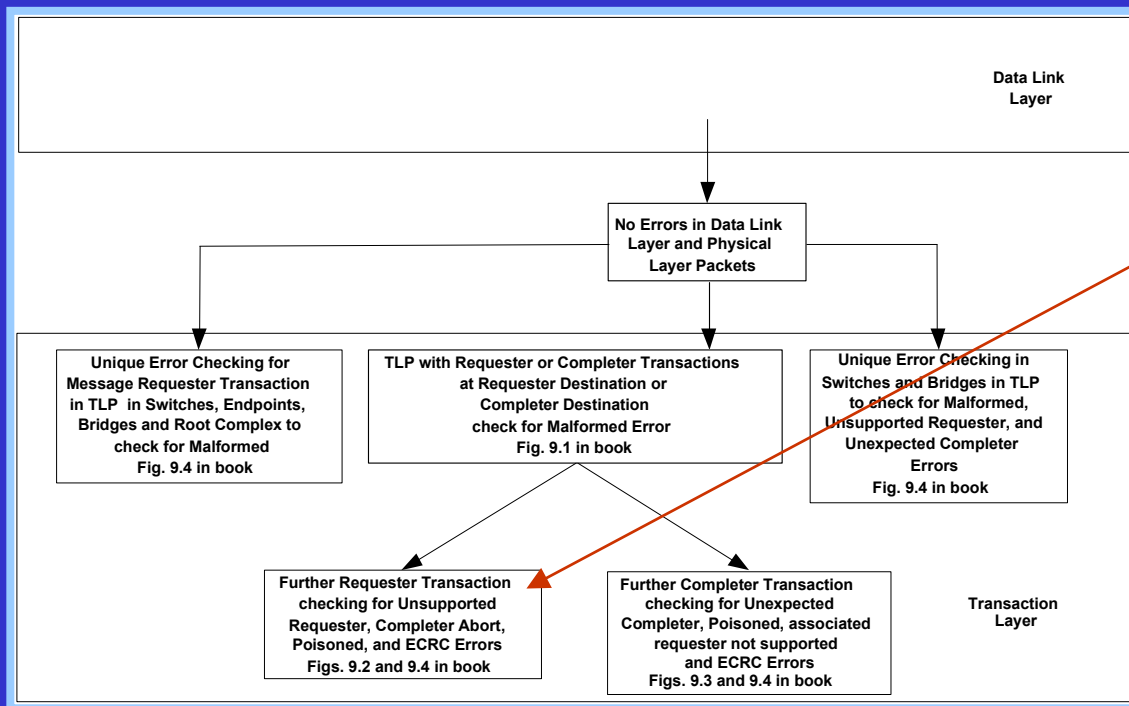
## Transaction Layer ... Basic Errors

- Assume there were no errors found in the Data Link or Physical Layers. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
- Within the Transaction Layer the Transaction Layer Packet (TLP) containing either requester or completer transactions must be first checked for the following:
  - Malformed: Determine if the form of the TLP is correct ...otherwise malformed error.
  - Type Defined:Determine if the TYPE field contains a valid TLP type ... otherwise malformed error.
  - These errors are reported as nonfatal or fatal via message requester transactions.



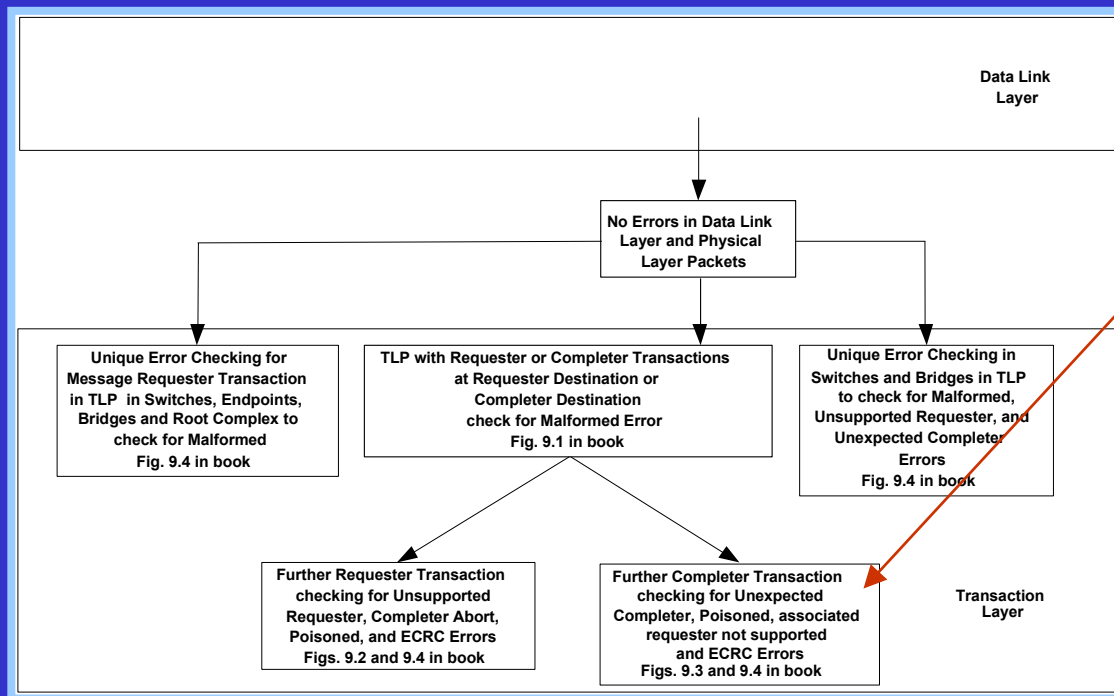
## Transaction Layer ... Basic Errors ... continued

- The TLP is “Further” checked specifically as a requester transaction. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - Unsupported: Determine if the TYPE of TLP (requester) is supported or if TLP is message the message code is defined ...otherwise unsupported requester error.
  - TLP processed and programming model correct ... otherwise completer abort error.
  - If TLP is successfully received (errors in the above two bullets did not occur), there are “Other Error” conditions that can cause completer abort or unsupported requester errors.



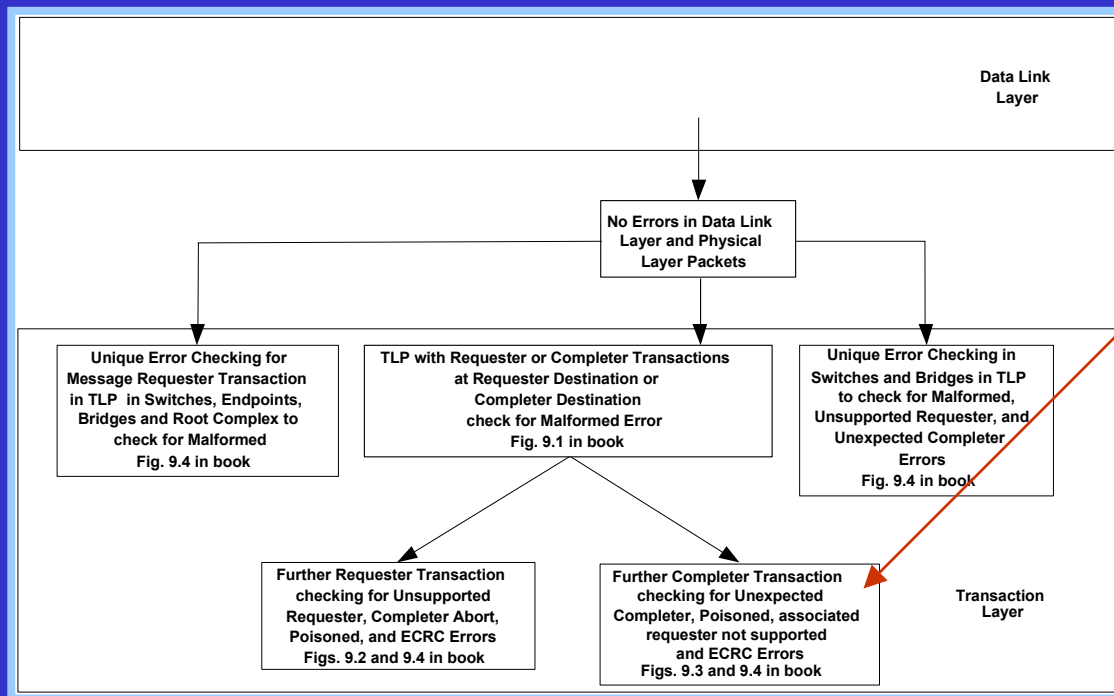
## Transaction Layer ... Basic Errors ... continued

- The TLP is then “Further” checked specifically as a requester transaction ... continued **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - If TLP is successfully received and errors in the above three bullets did not occur, there may still be errors related to a poisoned TLP and cyclic redundancy check.
  - These errors are reported as nonfatal or fatal via message requester transactions.



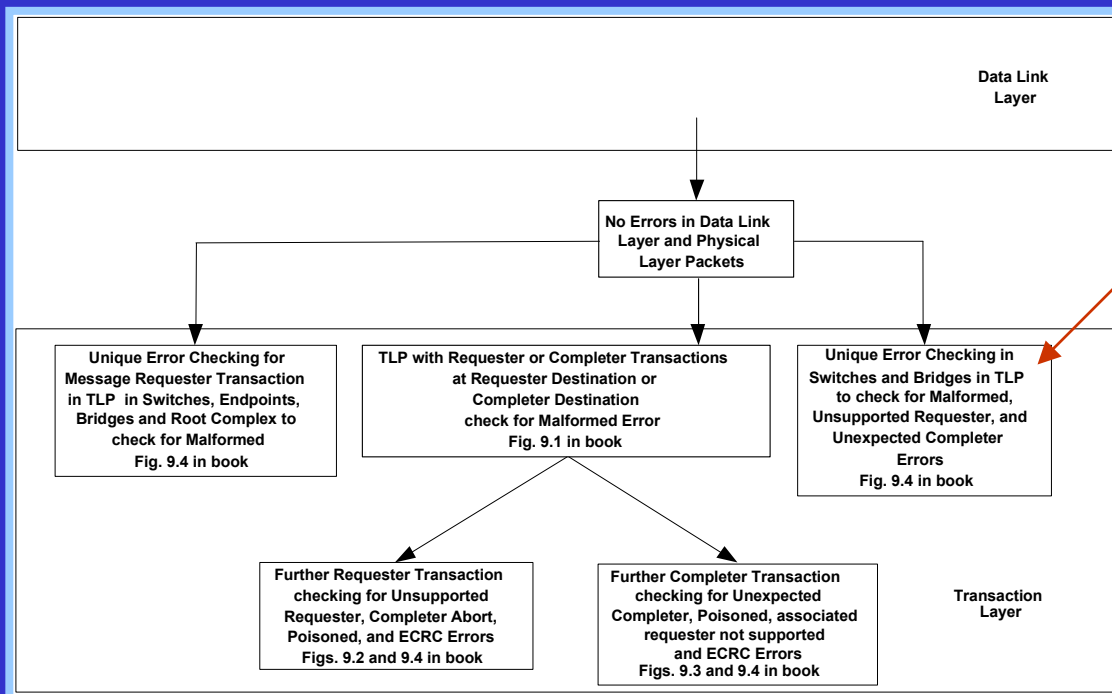
## Transaction Layer ... Basic Errors ... continued

- The TLP is then “Further” checked specifically as a completer transaction. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - TLP was expected for the associated requester transaction ... otherwise unexpected completer error occurred.
  - TLP is completer transaction indicates a successful status ... otherwise re-transmit requester transaction or report associated requester transaction as unsupported requester or completer abort error.



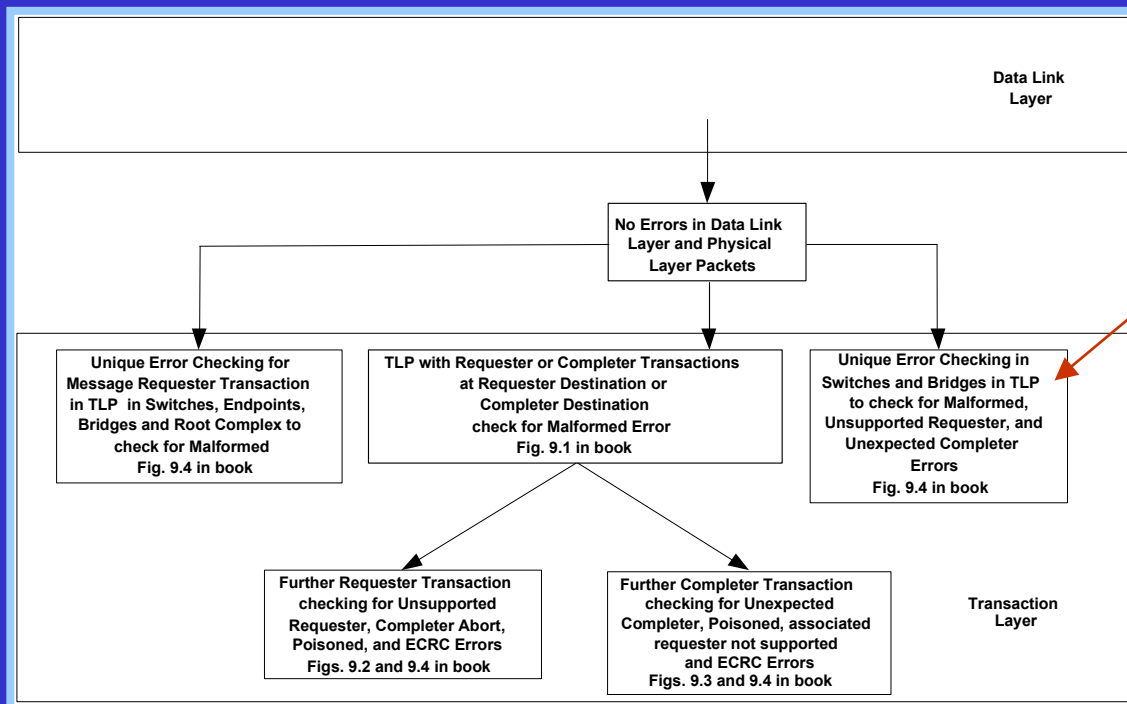
## Transaction Layer ... Basic Errors ... continued

- The TLP is then “Further” checked specifically as a completer transaction ... continued **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - If TLP is successfully received and the errors in the two bullets in previous slide did not occur. there may still be errors related to poisoned TLP and cyclic redundancy check.
  - These errors are reported as nonfatal or fatal via message requester transactions.



## Transaction Layer ... Basic Errors ... continued

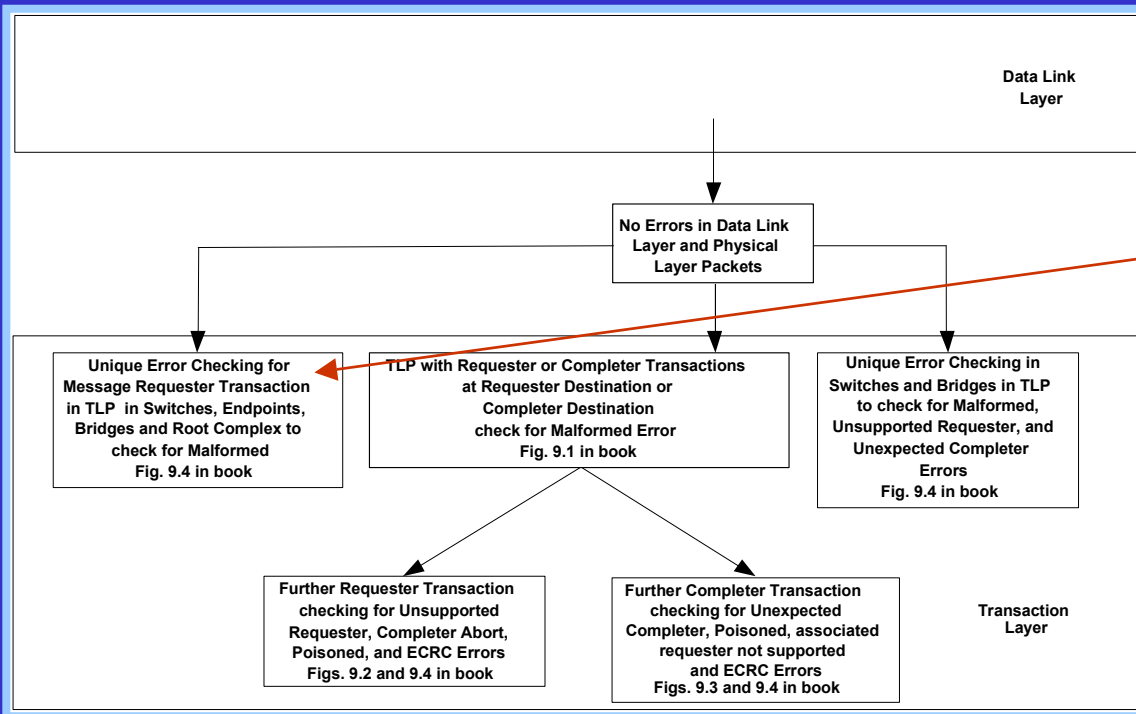
- TLPs porting through switches are checked for the following:  
**Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - If the address or routing information is not correct for porting of a TLP between ports, an unsupported requester error or unexpected completer error has occurred.
  - If the TC to VC mapping is not correct as defined by the switch, a malformed error has occurred.
  - These errors are reported as nonfatal or fatal via message requester transactions.



## Transaction Layer ... Basic Errors ... continued

- TLPs porting through bridges are checked for the following: **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - If the address range is not correct for the downstream PCI and PCI-X bus segments, or routing information is not correct for porting of a TLP between ports, an unsupported requester error or unexpected completer error has occurred.
  - These errors are reported as nonfatal or fatal via message requester transactions.



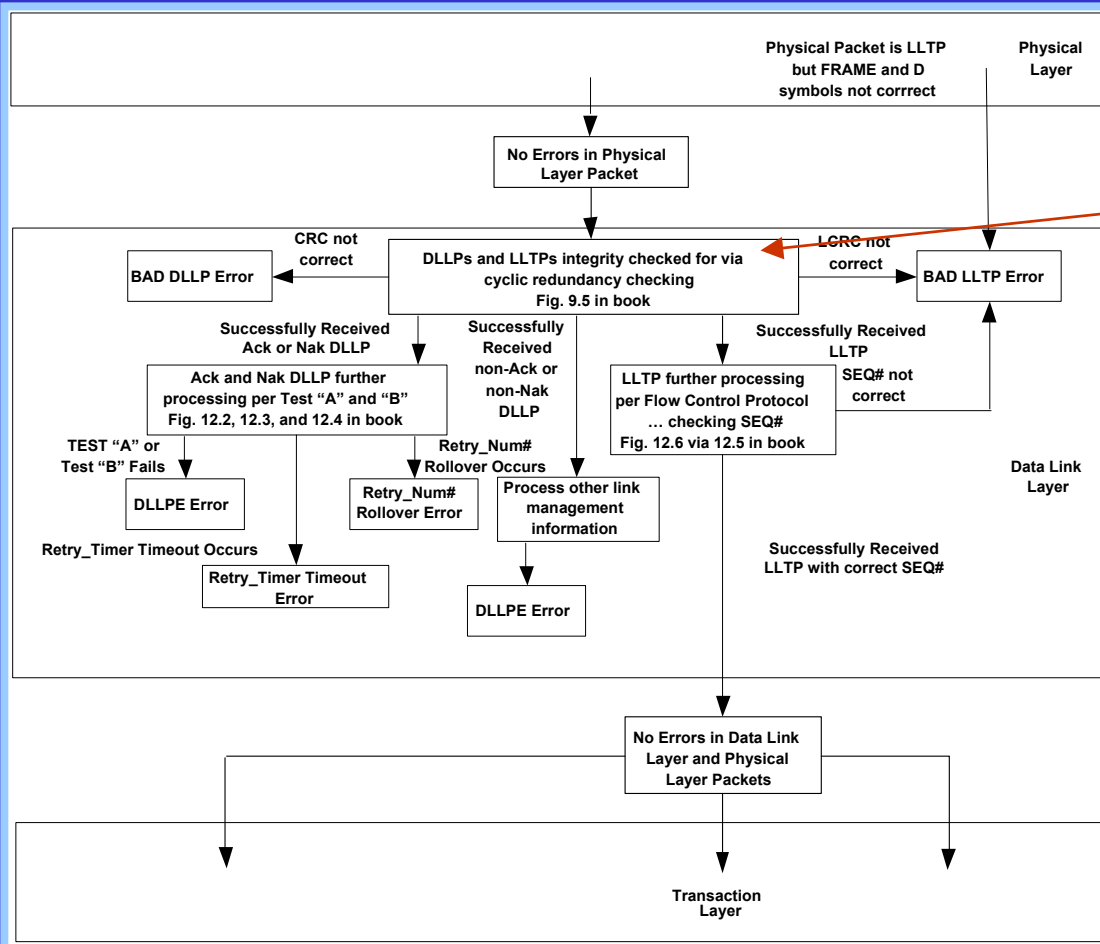


## Transaction Layer ... Basic Errors ... continued

- TLPs that contain message requester transactions the r field can optionally be checked by the switch, bridges, endpoints and Root Complex to determine if routing information is correct for the destination ...otherwise malformed error.
  - This errors is reported as nonfatal or fatal via message requester transactions.

### Other Transaction Layer Errors

- **Completion Timeout:** This error occurs if a completer transaction packet is not successfully returned to the requester source prior to the Completion Timeout timer expiring. The Completion Timeout timer is initialized and begins timing when the associated requester transaction is transmitted by the requester source.
  - This error is reported as nonfatal or fatal via message requester transactions.
- **Receiver Overflow:** Each TLPs encapsulated in LLTPs is only transmitted across the link if sufficient buffer space is available at the receiving port on the link. The receiving port can optionally check to see if the buffers have been overflowed by receiving too many TLPs.
  - This error is reported as nonfatal or fatal via message requester transactions.
- **Flow Control Protocol Error (FCPE):** During Flow Control Initialization of a link the minimal FCCs for each type of buffer are established. The transmitting port of the TLP encapsulated in the LLTP can optionally check to determine that the initial buffer space available is equal to or greater than the minimum, if not a FCPE is optionally reported.
  - This error is reported as nonfatal or fatal via message requester transactions.

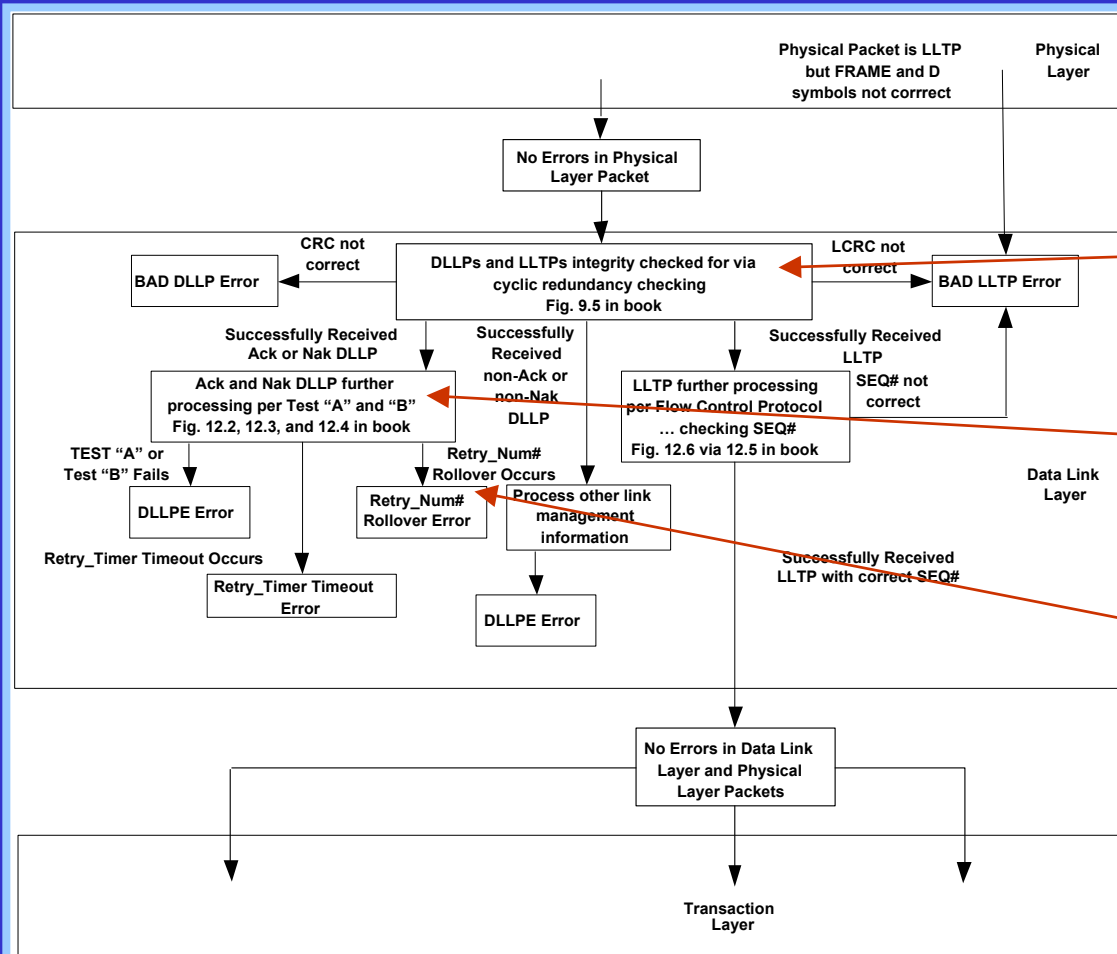


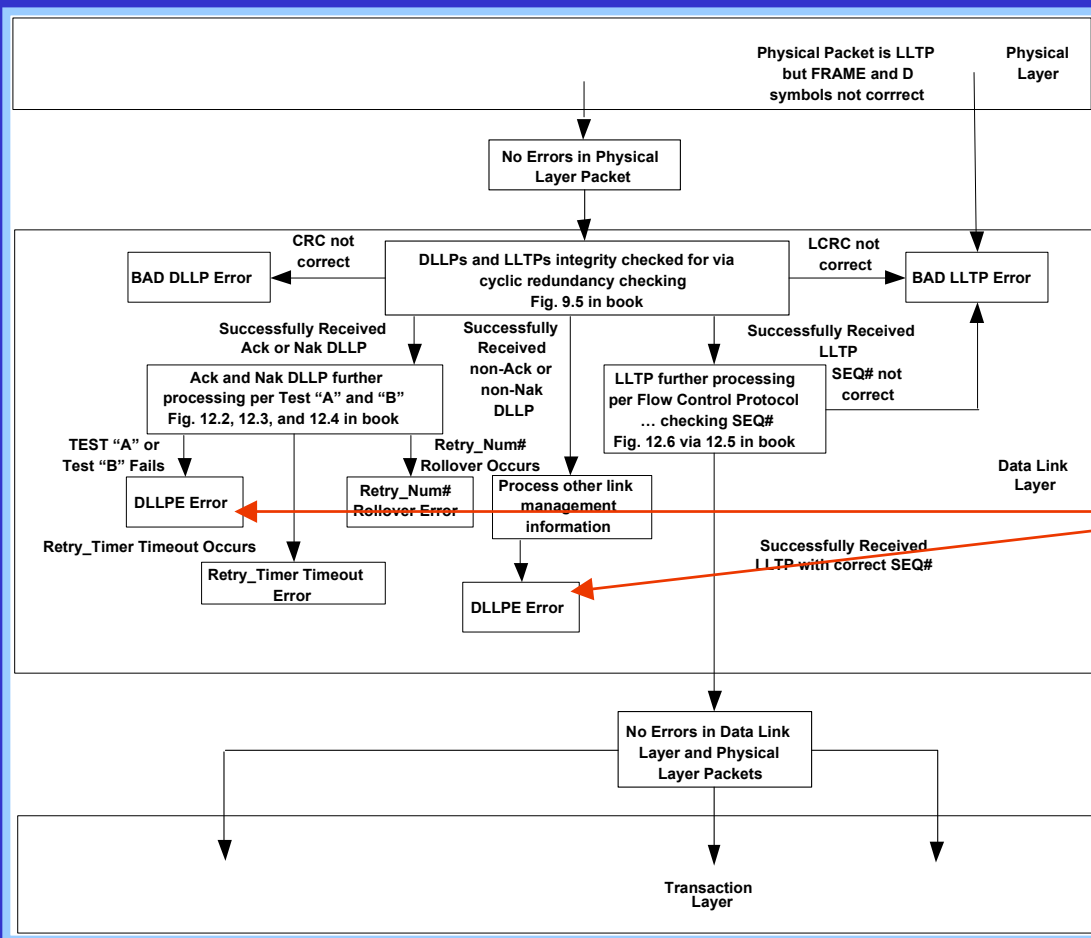
## Data Link Layer Errors

- Assumption is that there were no errors found in the Physical Layer.
- Within the Data Link Layer Transaction Layer the DLLP and LLTP are extracted from the Physical Packet and the following are checked: **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - If calculated cyclic redundancy check equals the CRC of the DLLP the DLLP has been successfully received ...otherwise a BAD DLLP error has occurred.
  - If calculated cyclic redundancy check equals the LCRC of the LLTP the LLTP has been successfully received ...otherwise a BAD LLTP error has occurred.
  - These errors are reported as correctable via message requester transactions.

## Data Link Layer Errors .. continued

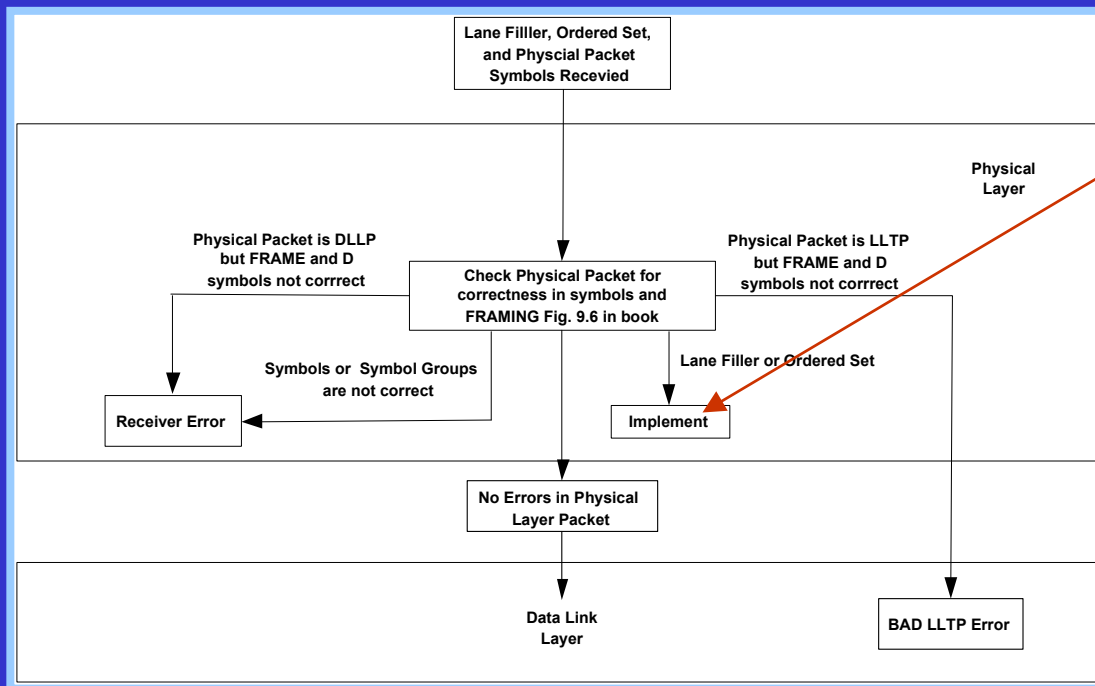
- Once it is determined that the DLLP or LLTP was successfully received there are other error considerations:  
**Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - LLTP: The SEQ# of the LLTPs received must be the next sequential number ...otherwise BAD LLTP error has occurred.
  - ACK or NAK DLLP: The LLTPs' transmitting port will retry the transmission until successful acknowledgement of receipt or until a Retry timer expires ...the later results in a Retry\_Timer Timeout error. If the LLTPs' transmitting port has too many Retry\_Timer Timeouts, the port reports a Retry\_Num#\_Rollover error. All three of these errors are part of the Flow Control protocol of Chapter 12 These errors are reported as correctable via message requester transactions.





## Data Link Layer Errors ... continued

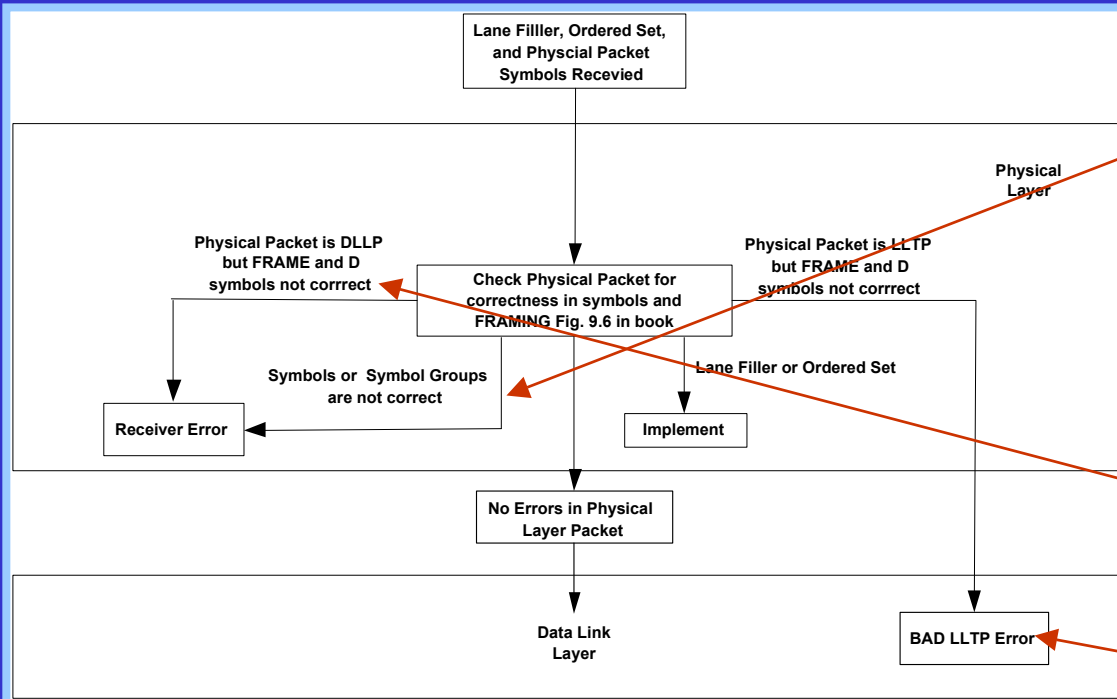
- Once it is determined that the DLLP or LLTP was successfully received there are other error considerations: .... Continued  
**Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - For Flow Control Initialization or when the value of the AckNakSEQ in the DLLPs does not equal an unacknowledged LLTP in the retry buffer an optionally checked DLLPE (Data Link Layer Packet Error) occurs.
  - This error is reported as nonfatal or fatal via message requester transactions.



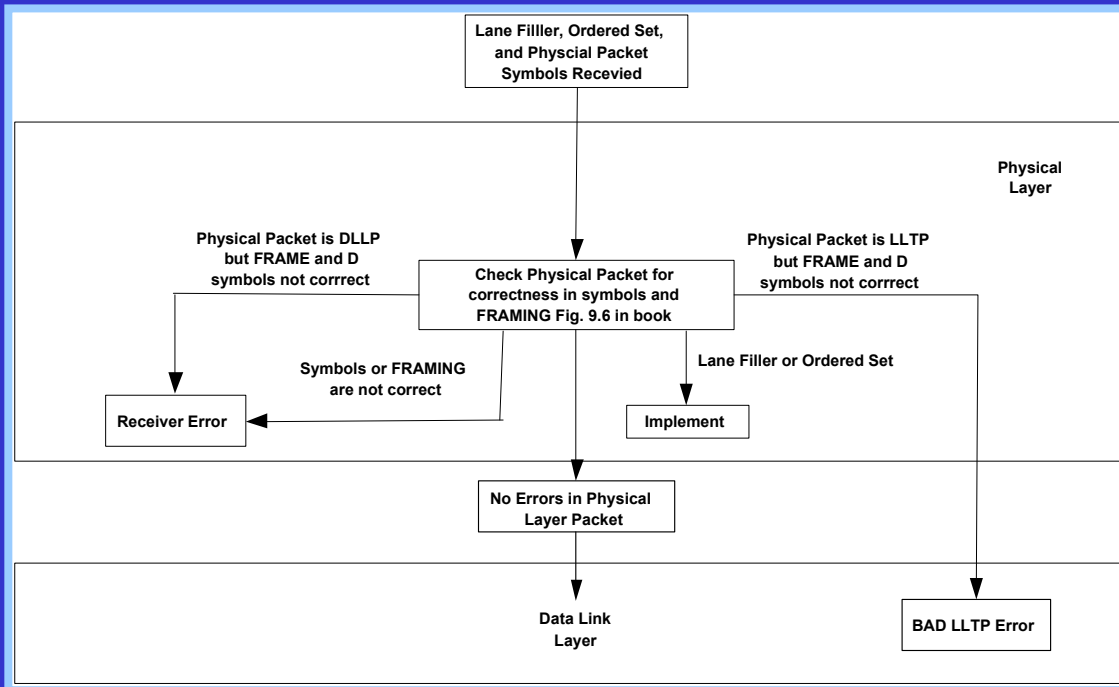
## Physical Layer Errors

- The Physical Packets are received intermixed among lane fillers and Ordered Sets. Assuming these are received without error. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - The lane fillers are either PAD K or IDLE D symbols. PAD K symbols inserted in the lanes to insure the LLTPs and DLLPs are aligned to LANE# 0. The IDLE D symbols ensure retention of bit and symbol lock at the receiver when no LLTPs or DLLPs are ready for transmission.
  - The Ordered Sets are used for link state transactions of the LSSTM and associated links to accomplish different tasks.

## Physical Layer Errors ... continued



- There may be a Receiver Error in the symbol stream for FRAMING of LLTPs, lane fillers, or Ordered Sets. If there is an error the symbols are discarded and Receiver Error is reported. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
- The portion of the stream of symbols that are Physical Packets are further processed for errors as follows:
  - The combination of FRAMING and D symbols seems to indicate a DLLP but are not correct ... Receiver Error is reported.
  - The FRAMING and D symbols are correct and indicate LLTP but the LLTP has an error ... a BAD LLTP error is reported.
- The Receiver Error is correctable and the BAD LLTP error is nonfatal or fatal. Both errors are reported via message requester transactions.



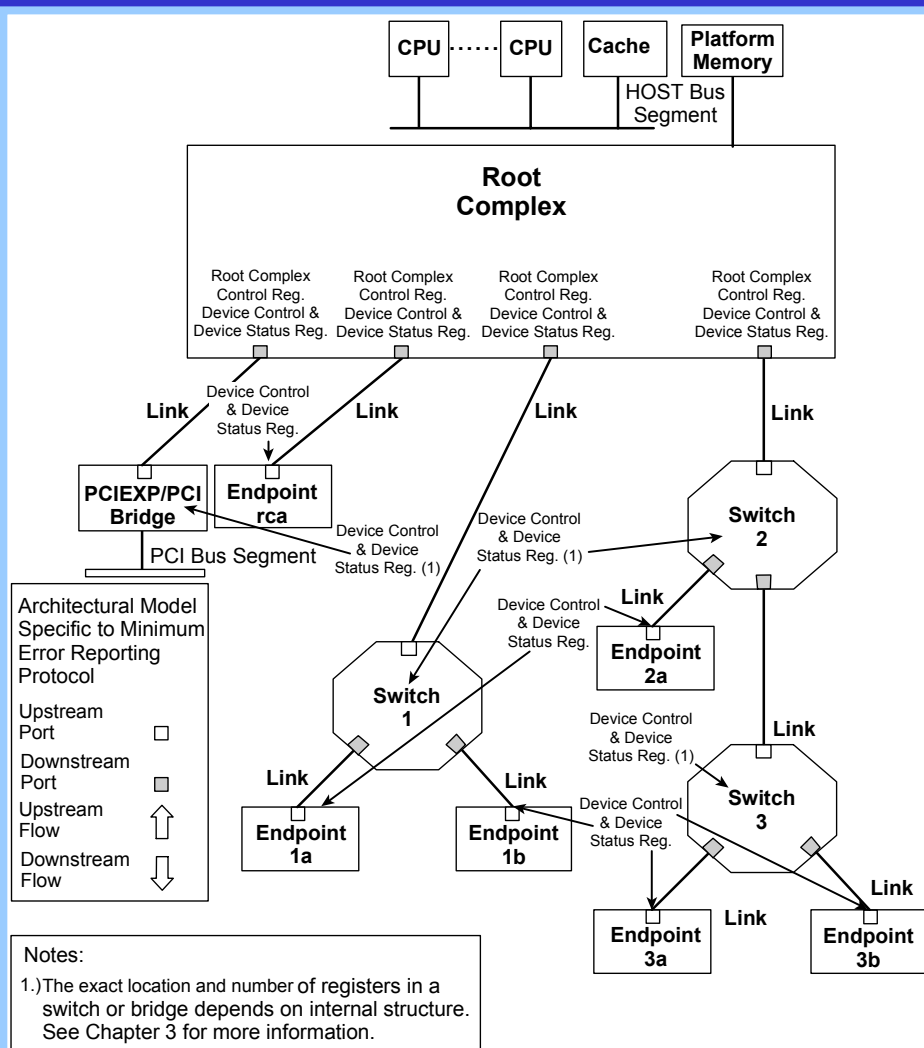
### Physical Layer Errors ... continued

- Training Error: A PCI Express device may optionally check for any violation of the Link Training protocol and report them as report Training Errors accordingly. Elements of Link Training are also executed in the Link Retraining during the Recovery link sub-states .
- This error is reported as nonfatal or fatal via message requester transactions.



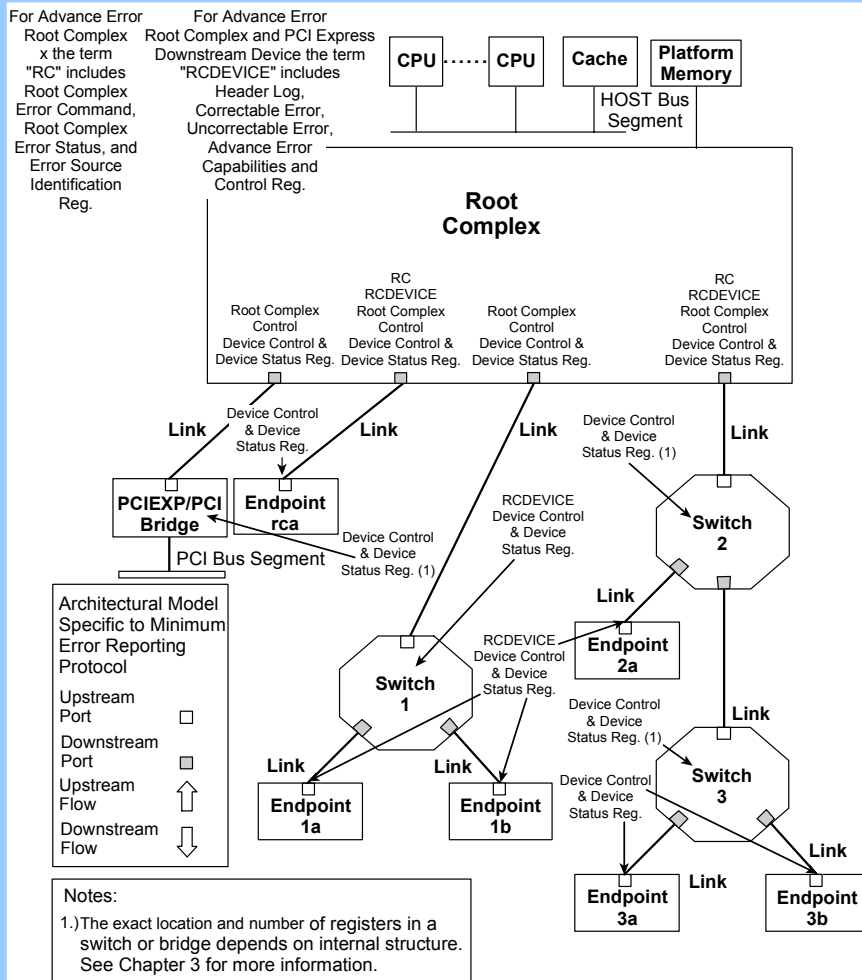
## Error Reporting and Logging

- Any error that is detected in the Transaction, Data Link, or Physical Layer is reported to the Root Complex.
- To report the error the PCI Express device that detected the error transmits a message requester transaction packet.
- The errors defined for each layer are defined as one of three types of errors as follows:
  - Transaction Layer
    - All errors from this layer can be reported as nonfatal and fatal.
  - Data Link Layer
    - BAD LLTP and BAD DLLP are defined as correctable.
    - Retry\_Timer Timeout and Retry\_NUM# Rollover are defined as correctable.
    - DLLPE is defined as nonfatal and fatal.
  - Physical Layer
    - Receiver Error is defined as correctable.
    - Training Error is defined as nonfatal and fatal.
- A specific message transaction packet is defined for each of the three types of errors:
  - Correctable
  - Fatal
  - Nonfatal
- The message error requester transaction packet is sourced by the PCI Express device that detected the error if the appropriate bit is enabled. The destination of the message error requester transaction packet is the Root Complex and is processed on two levels: Minimum Error Reporting and Advance Error Reporting.



## Error Reporting and Logging ... continued

- Both Minimum and Advance Error Reporting protocols rely on the message error transaction packets to inform the Root Complex of the error. In both protocols the transmission of the packet and the recognition of the packets by the Root Complex may or may not be enabled.
- When one of the three types of errors (correctable, nonfatal, or fatal) the appropriate message error requester transaction packet is transmitted if the associated bit of the function's Device Control register is set = 1b. Also, the appropriate bits are set = 1b in the function's Device Status register independent whether a message error requester packet is transmitted or not. A system error is generated upon receipt of the message error transaction packet if BIT# 0, BIT# 1, or BIT# 2 in Root Complex Control register is set = 1b. The method by which the system error is generated in the Root Complex to the HOST bus segment is implementation-specific.



## Error Reporting and Logging ... continued

- If the message error requester transaction packet is transmitted and recognition is enabled, the two protocol process the error as follows:
  - Minimum Error Reporting Protocol alerts the Error Software via a system error defined per the PCI protocol.
  - Advance Error Reporting can use system error or interrupt instead.
- Once Error software has been alerted there has been an error there are two levels of registers that can be checked depending on the error reporting level.
  - Minimum Error Reporting provide no additional information about the error and thus is implemented by the minimum number of configuration block registers and only support generating system error
  - Advance Error Reporting provides the option to alert Error software by an interrupt instead of system error. In addition the Advance Error Reporting protocol also implements the following configuration registers in the downstream PCI Express devices. Header Log, Correctable Error, Uncorrectable Error, and Advance.

# The Complete PCI Express Reference Topic Group 3 Tutorial

Disclaimer: Intel, Research Tech Inc. and the authors of this tutorial make no warranty for the accuracy or use of the information No direct or indirect liability is assumed and the right to change any information without notice is retained.

## Design Tools for PCI Express

The PCI Express specification is not organized by clear design topics, does not provide sufficient details to easily master PCI Express, and does not provide in depth illustrations and flowcharts to assist in designing components. Intel recognized that simply summarizing or re-wording the specification as typically done in the other design tools in the industry is insufficient ... more extensive information is are needed.

To provide designers with PCI Express design and implementation information that is easy to master and detailed enough to assist in correct design, two design tools are available:

Six Detailed Tutorials and a new and exhaustively detailed design book.

These design tools focus on Six Quick and Easy Topic Groups which simplify the mastery of PCI Express. They save a designer weeks of trying to unravel the specification and provide the assurance of correct design implementation the first time.

## Design Tools for PCI Express

### The “Book”

The primary design tool that provides total design mastery is *The Complete PCI Express Reference* book written by Edward Solari and Brad Congdon and published by Intel ... referred to as the “Book”.

The Book provides the complete and extensive narrative of detailed figures (over 250), detailed design flow charts, and exhaustive tables for the complete understanding and design assistance in over 1000 pages. The Book can be ordered at [www.amazon.com](http://www.amazon.com) ... ISBN # 0971786194.

### Detailed Tutorials

Six free **Detailed Tutorials** ... One self paced tutorial for each of the Six Quick and Easy Topic Groups. Each introduces PCI Express information with a narrative that complements detailed figures, flow charts, and tables for each specific Topic Group from the Book. The six free **Detailed Tutorials** are available at [www.intel.com/intelpress/pciexpresscomplete](http://www.intel.com/intelpress/pciexpresscomplete).

### This Detailed Tutorial for Topic Group 3

Detailed Tutorial: *Transaction Ordering and Flow Control Part 1 and 2 Protocols*  
References in the Book: *Chapters 10 to 12*

## PCI Express in Six Topic Groups

### Topic Group 1

**Tutgroup1: *Platform Architecture and Accessing of Resources within Architecture***

**References in the Book: *Chapters 1 to 4***

### Topic Group 2

**Tutgroup2 : *Packets' and Layers' Specifics and Errors***

**References in the Book: *Chapters 5 to 9***

### Topic Group 3

**Tutgroup3 : *Transaction Ordering and Flow Control Part 1 and 2 Protocols***

**References in the Book: *Chapters 10 to 12***

### Topic Group 4

**Tutgroup4 : *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

**References in the Book: *Chapters 13 to 17***

### Topic Group 5

**Tutgroup5 : *Other Hardware Topics***

**References in the Book: *Chapters 18 to 21***

### Topic Group 6

**Tutgroup6 : *Software Considerations***

**References in the Book: *Chapters 22 to 24***

# Transaction Ordering and Flow Control

## Chapters 10 to 12

### Topic Group 3

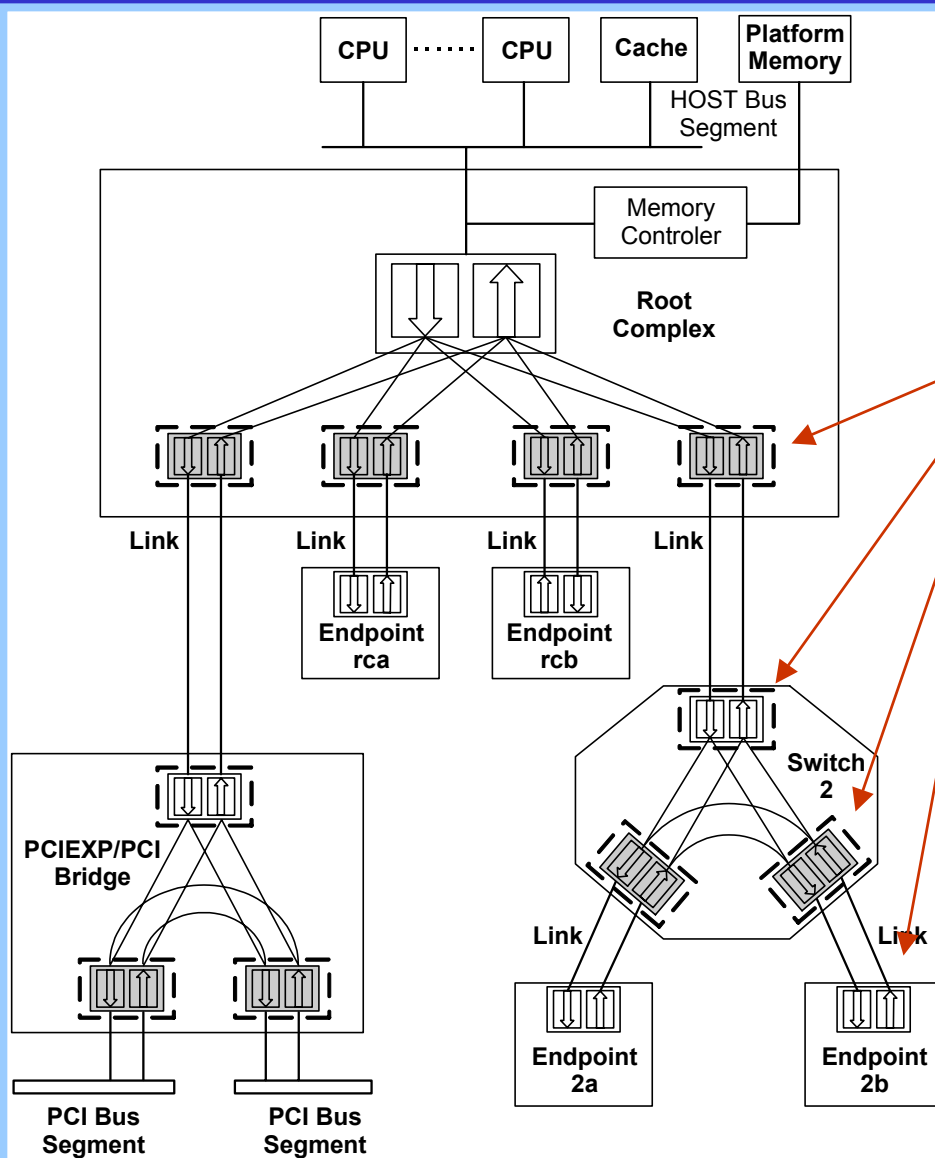
The requester and completer transactions between PCI Express devices are flowing through the platform per specific protocols.

**Summary:** The requester and complete transactions that are interact between PCI Express devices do so via packets. The packets of particular importance are the Transaction Layer Packets (TLPs). Throughout the PCI Express platform there are buffers which merge TLPs from different sources and determine the next TLP to be transmitted onto the link.. The protocols that govern the flow of TLPs and operation of these a buffers are Transaction Ordering and Flow Control.



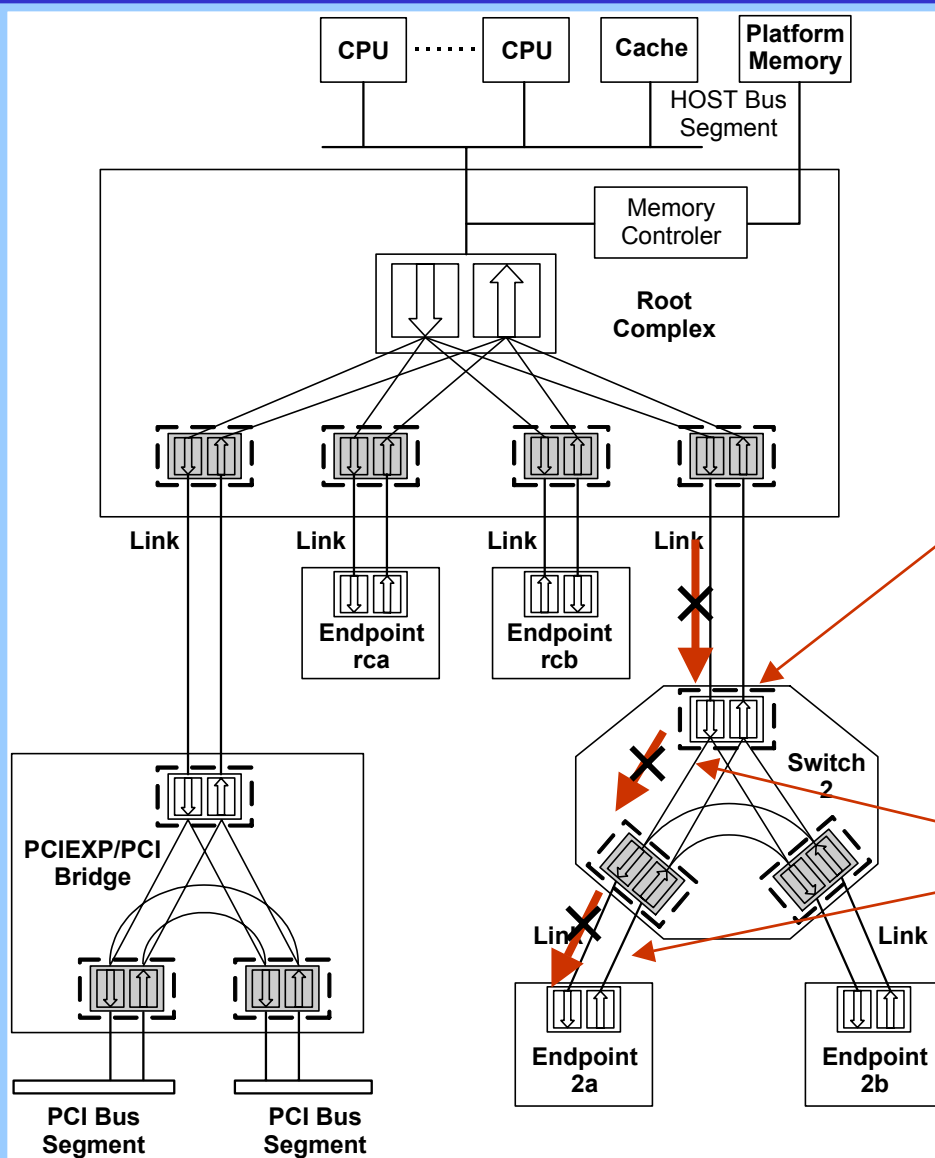
# Chapter 10

## Transaction Ordering



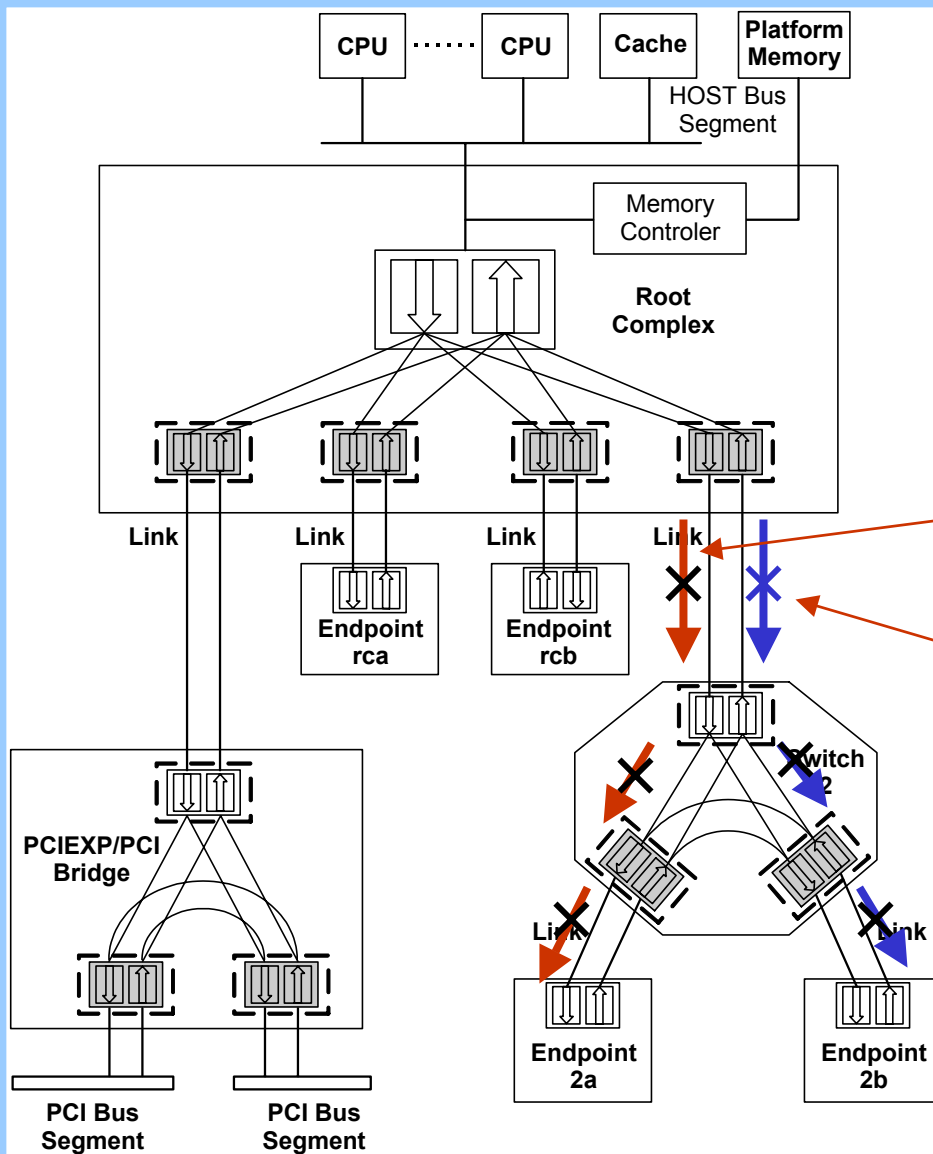
## Transaction Ordering

- At each transmitting port of a PCI Express device a buffer contains the TLPs to be transmitted over the link.
- The Requester/Completer protocol implemented by PCI Express means that many requester transactions packets have to be transmitted across the PCI Express platform. Consequently; the completion of many transactions are pending the receipt of the associated completer transactions.
- The flow of TLPs is from buffer to buffer. At each buffer the TLPs are merged with other TLPs to be transmitted onto the link.
- The TLPs flowing downstream through the buffers and links are independent of the TLPs flowing upstream. That is TLPs flowing in opposite directions do not interact.



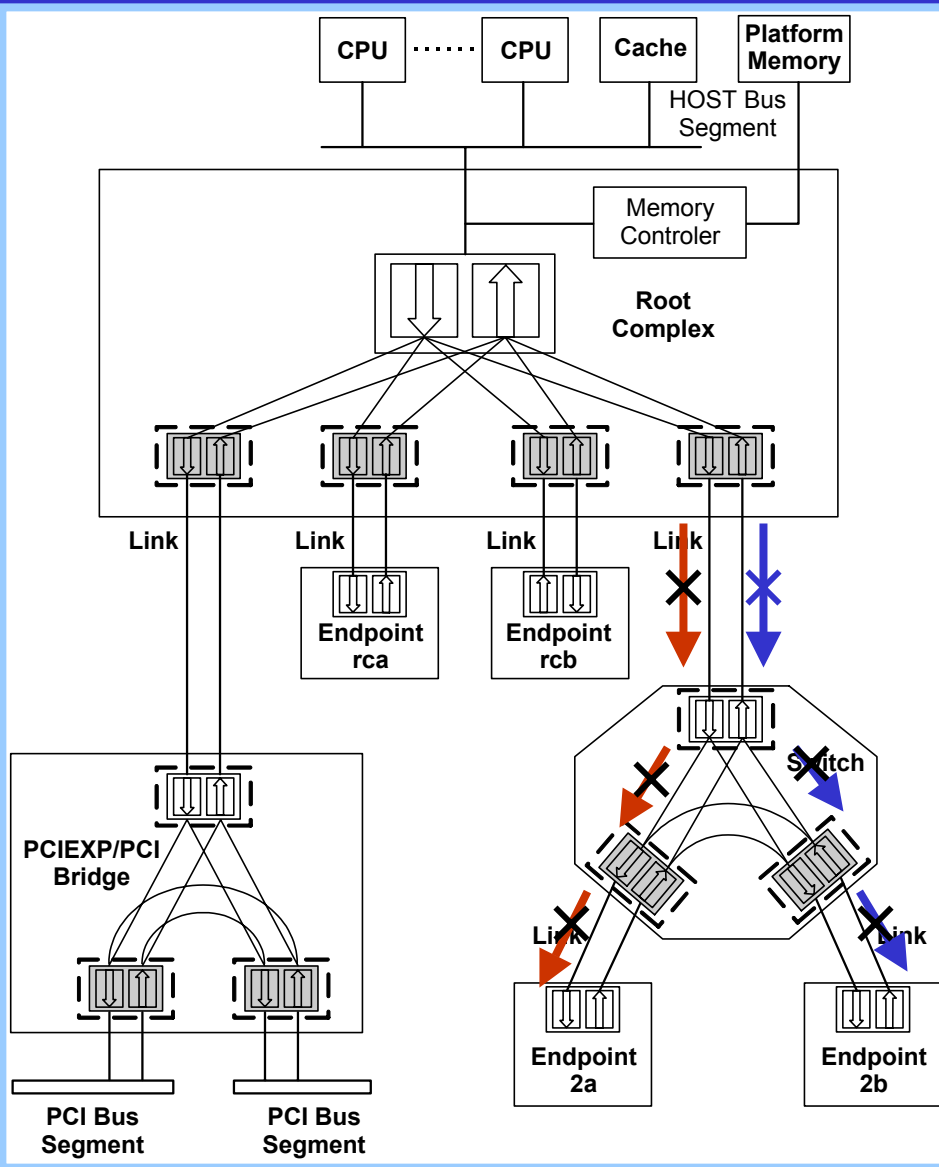
## Transaction Ordering ... continued

- The flow of TLPs in the same direction are represent a mix of requester and completer transactions that are associated with multiple PCI Express devices. The forward movement of TLPs from a buffer (transmitter) to the link requires that buffer space is available at the other end of the link (receiver).
  - If buffer space is not available at a specific receiver on a link, the TLP that could be transmitted from the buffer on the other end of the link can not be transmitted.
  - There can be a ripple effect that prevents TLPs from the other ports of a switch to port to the transmitting ports of the switch all the way back to the source.



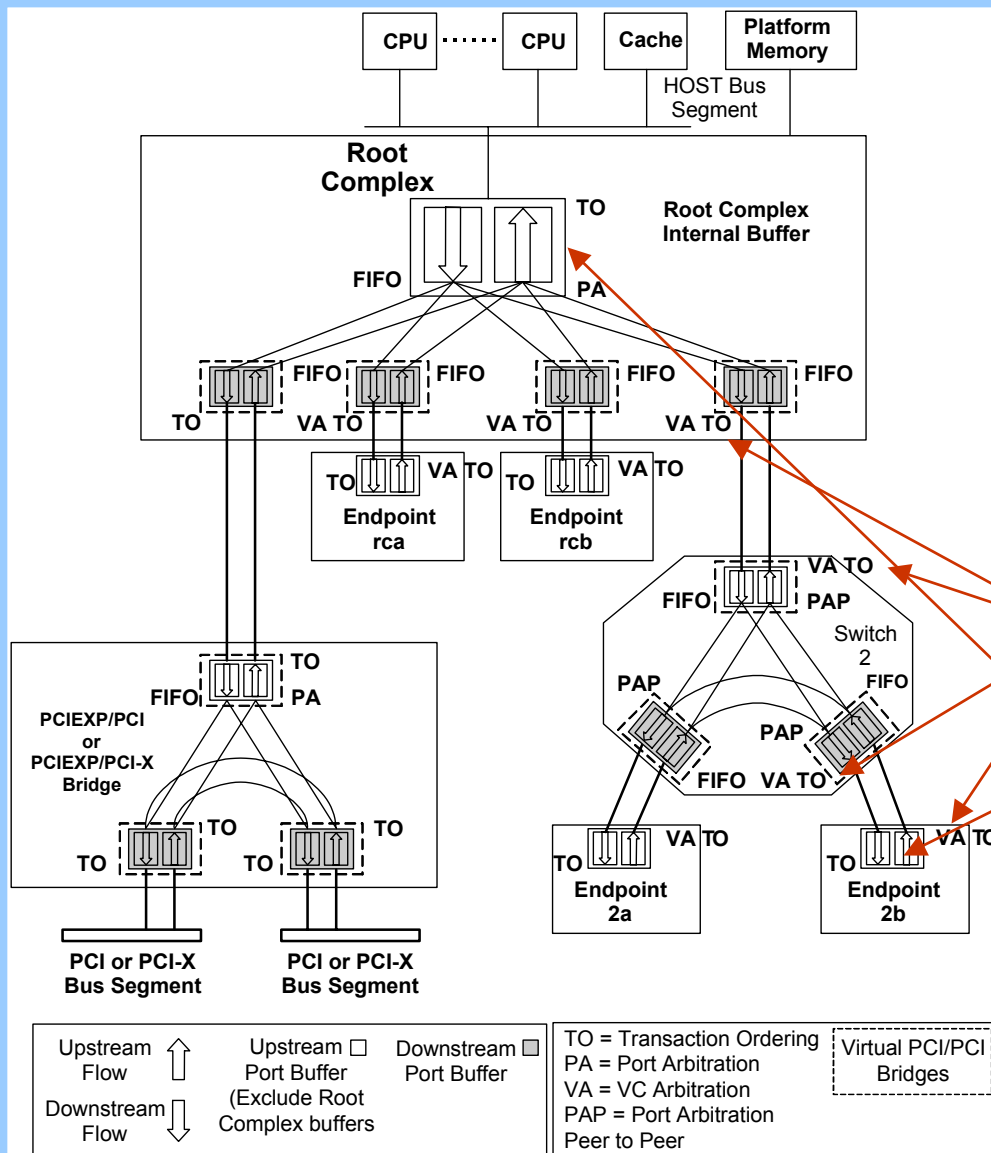
## Transaction Ordering ... continued

- If all the buffers are structured as true FIFOs, it is possible that TLPs associated with one PCI Express device is being blocked by the inability of TLPs associated with another PCI Express to move forward.
  - For example, assume all the TLPS representing requester transactions packet are backed up between the Root Complex (Source) and endpoint 2a (destination).
  - If the downstream port of the Root Complex implemented with a FIFO it is possible that a completer transaction from the Root Complex (source) can not proceed forward to endpoint 2b (destination) due to the backup at the FIFO.
  - Consequently endpoint 2b can not complete its transaction because of a backup at the Root Complex caused by a backup due to endpoint 2a.



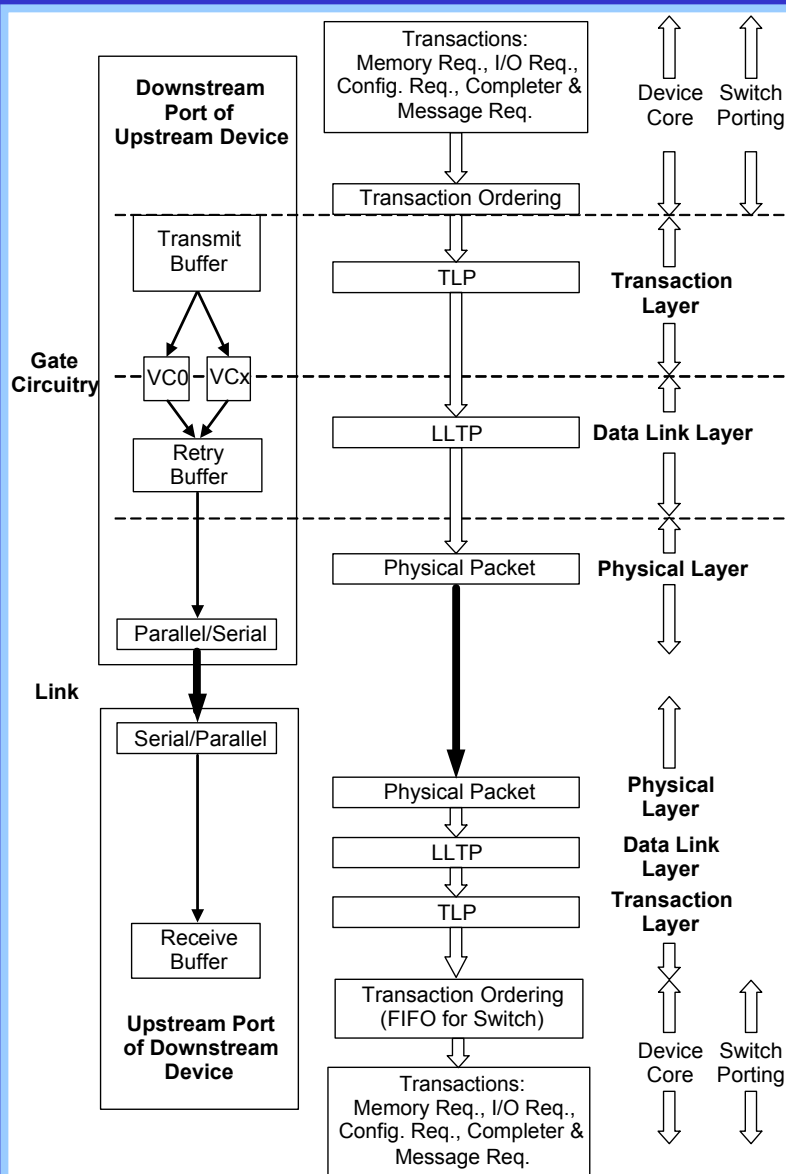
### Transaction Ordering ... continued

- If the inability of TLPs to not more forward (transmitted from a buffer onto the link) is never resolved, a deadlock condition occurs. If the inability is short lived it is livelock condition.
- The purpose of the Transaction Ordering protocol is to prevent livelock and dead lock conditions.



## Transaction Ordering ... continued

- In order to address livelock and deadlock conditions, the Transaction Order protocol is applied to the transmitting ports.
- For purposes of simplifying the discussion the buffer model applied to the receiving ports is that of a FIFO. The strong ordering of a FIFO for TLP transmission may not be applied in all cases. See the Book for more information.
- The Transaction Ordering protocol (TO) is applied to all other buffers at the transmitting ports.
- TO can be applied to other buffers not associated with a PCI Express transmitting port connected to a link. This is implementation specific.
- The other elements relative to buffer design (VA, PA, and PAP) will be discussed in later slides.



## Transaction Ordering ... continued

- The Transaction Ordering (TO) protocol permits hardware to determine which TLPs can be transmitted next from a buffer instead of the strict transmission priority of a FIFO.
  - Essentially, if one TLP can not be transmitted due to receiver buffer space not being available, another TLP that fits the available buffer space at the receiver can be transmitted.
  - Thus, some of the TLPs are allowed to make forward process through the buffer.
- The forward movement of TLPs and thus the implementation of TO occurs at two levels
  - The interface between the PCI Express device core and the Transaction Layer (Gate circuitry) must consider TO. Once established the TO must be retained as the TLP is processed through the layers.
  - In the case of switches, TLPs are ported from the switches' receiving ports through the Address Mapping & TC to VC Mapping Module and must also consider TO as TLPs are ported to the switches' transmitting ports via Gate circuitry and the layers.

Subsequent Transaction vvv	Reference Transaction				
	Requester (Posted) Memory Write or Message (U) > (D)	Requester Memory Read (U) > (D)	Requester I/O or Config. Write (U) > (D)	Completer for Read (U) > (D)	Completer for Write (U) > (D)
Requester Transaction: (Posted) Memory Write or Message (U) > (D)	Cannot pass	Must pass	Must pass	Must pass	Must pass
Requester Transaction: Memory Read (U) > (D)	Cannot pass	May pass	May pass	May pass	May pass
Requester Transaction: I/O or Config. Write (U) > (D)	Cannot pass	May pass	May pass	May pass	May pass
Completer Transaction for Read (U) > (D)	Cannot pass	Must pass	Must pass	May pass	May pass
Completer Transaction for Write (U) > (D)	May pass	Must pass	Must pass	May pass	May pass

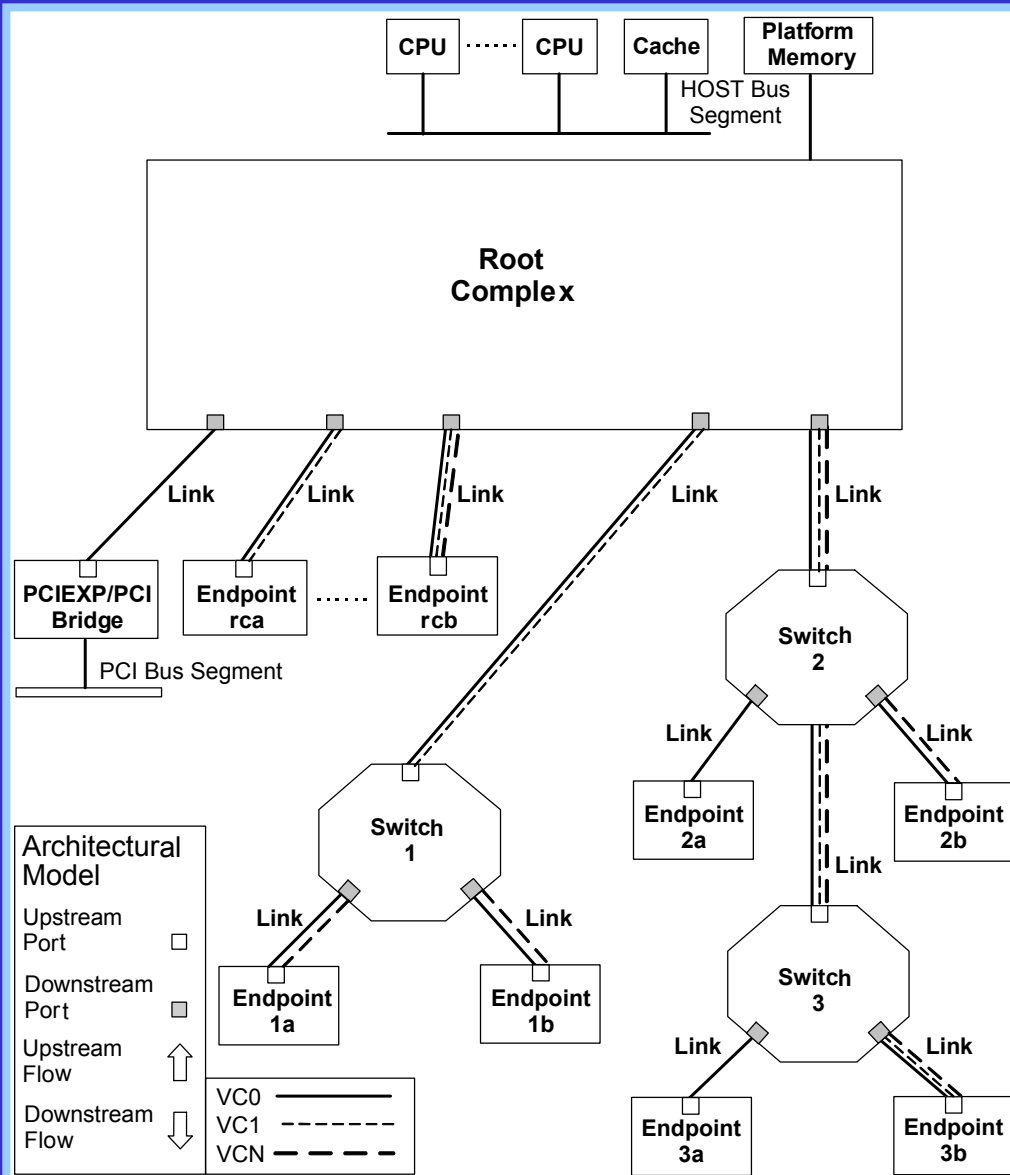
## Transaction Ordering ... continued

- All the TO does is permit TLPs to be processed for transmission ahead of other TLPs that were previously posted in a buffer.
- To prevent software execution error, the TLPs that can be processed and transmitted before others is dependent on other TLPs posted.
  - Example 1, A TLP consisting of a memory write requester transaction must be processed (pass) before other TLPs in the buffer (Reference Transaction) except for those also containing posted memory write requester transactions.
  - Example 2, A TLP consisting of a completer transaction associated with a write requester transaction may or may not be processed (pass) before other TLPs in the buffer (Reference Transaction). The Gate circuitry can make a real time determination either way.
- The TO only applies to TLPs assigned to the SAME Traffic Class number.
- Within the TO protocol there are two levels: PCI and PCI-X compatible. The level applied is defined by the contents in the ATTRIBUTE bits in the Header field of the TLP. See the Book for more information.
- Note: This is a sample of one of two TO tables detailed in the Book, see the Book for full information.



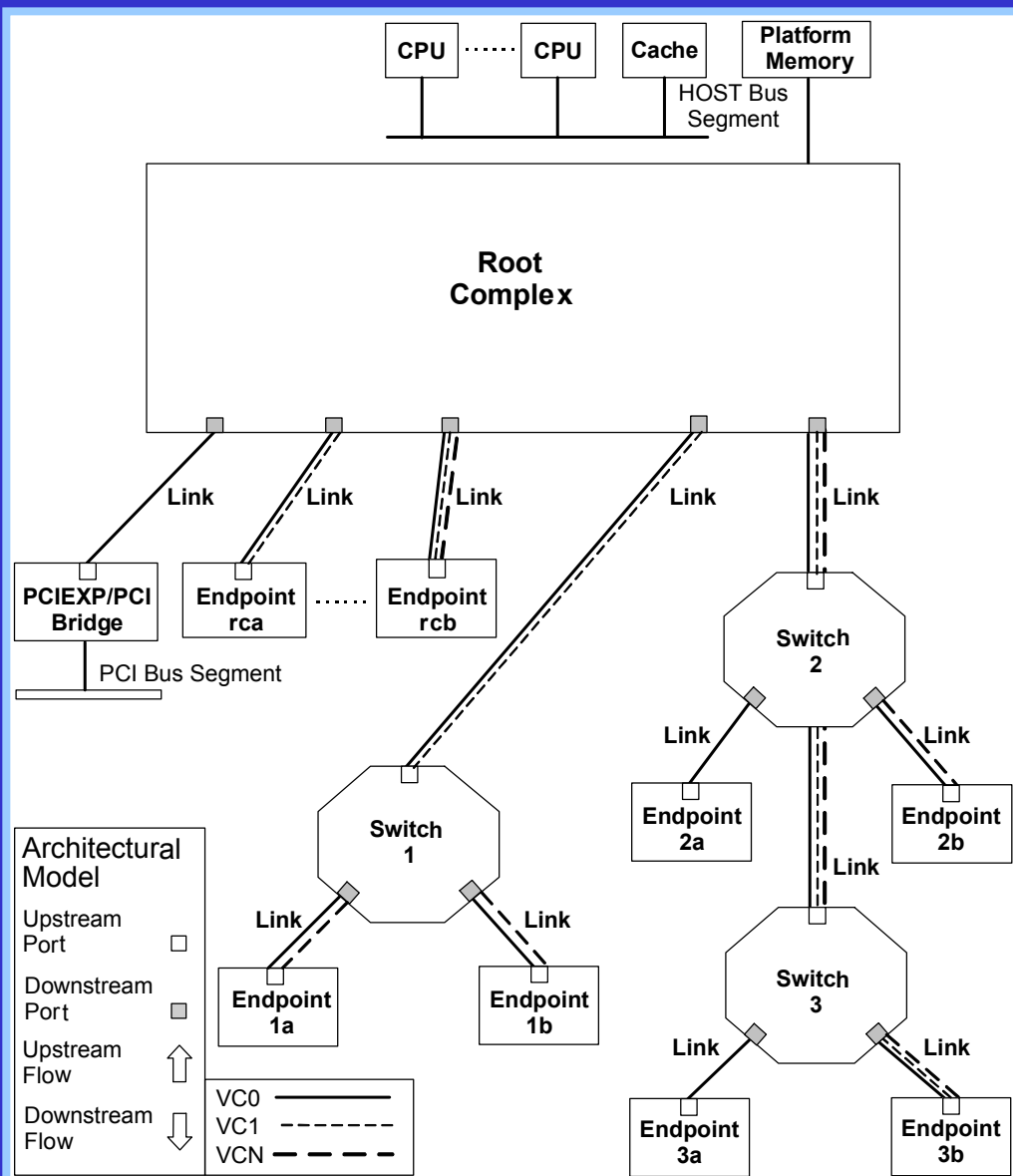
# Chapter 11

## Flow Control Protocol Part 1



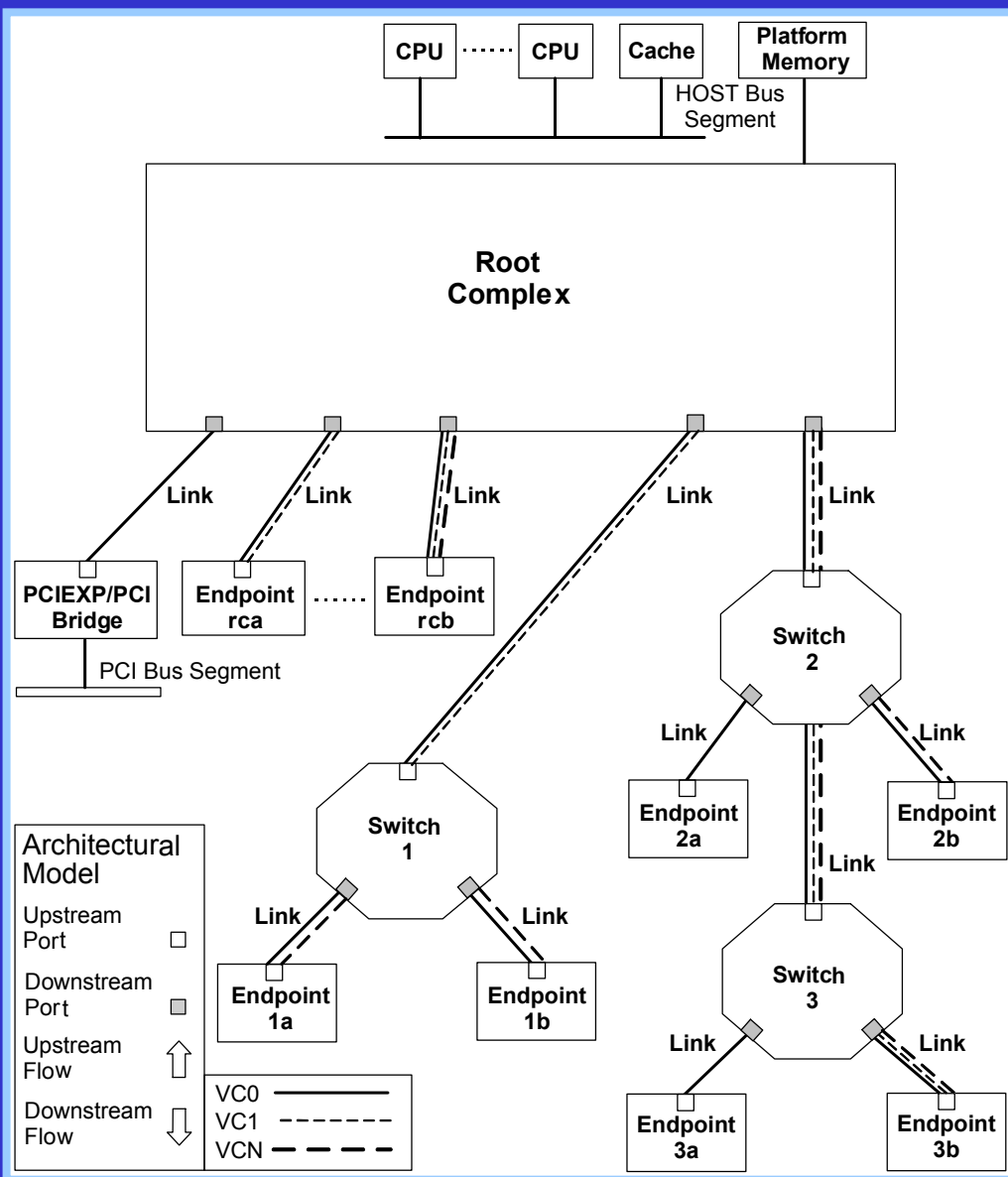
## Flow Control Introduction

- The previous slides detailed the Transaction Ordering protocol which defines how to avoid livelocks and deadlocks. This protocol aids in the performance of the platform, but does not provide a method to fine tune elements of the PCI Express fabric of the platform
- As discussed in earlier slides the links between the PCI Express devices consist of multiple lanes. The different number of lanes per each link defines a different maximum bandwidth of each link.
- The maximum bandwidth only defines the throughput of the TLPs encapsulated in LLTPs and contained in the Physical Packets. At each transmitting port a buffer contains all of the TLPs to be transmitted onto the link.



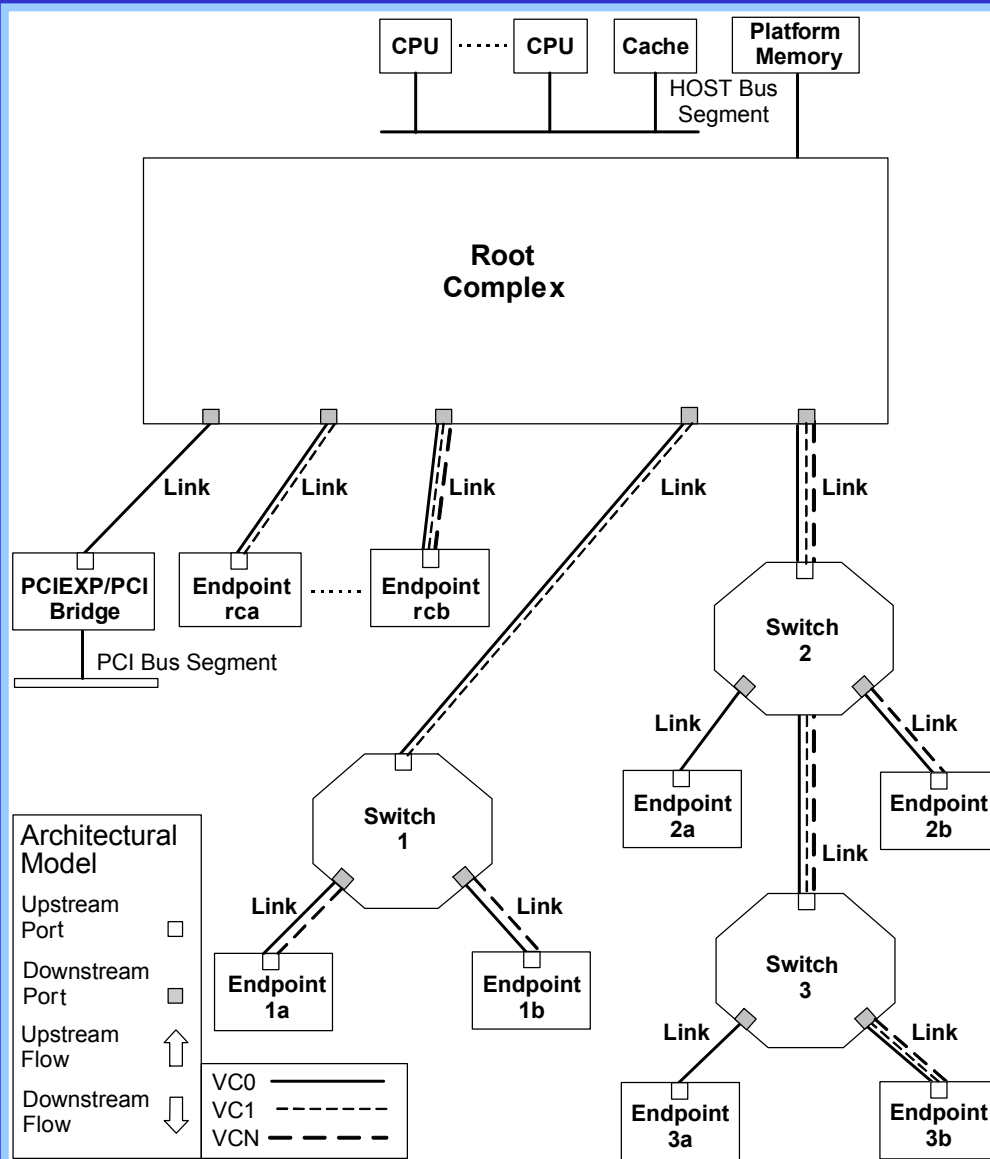
## Flow Control Introduction ... continued

- The simplest approach is to honor the Transaction Ordering protocol for livelock and deadlock considerations, and provide no other considerations relative to transmission priority.
- The simplest approach does not allow software to fine tune the PCI Express fabric to prioritize link bandwidth to TLPs associated with specific transactions. This ability to fine tune the priority for link bandwidth is defined a Quality of Service. An example of Q of S is isochronous transactions which requires known latency, bandwidth performance, etc..



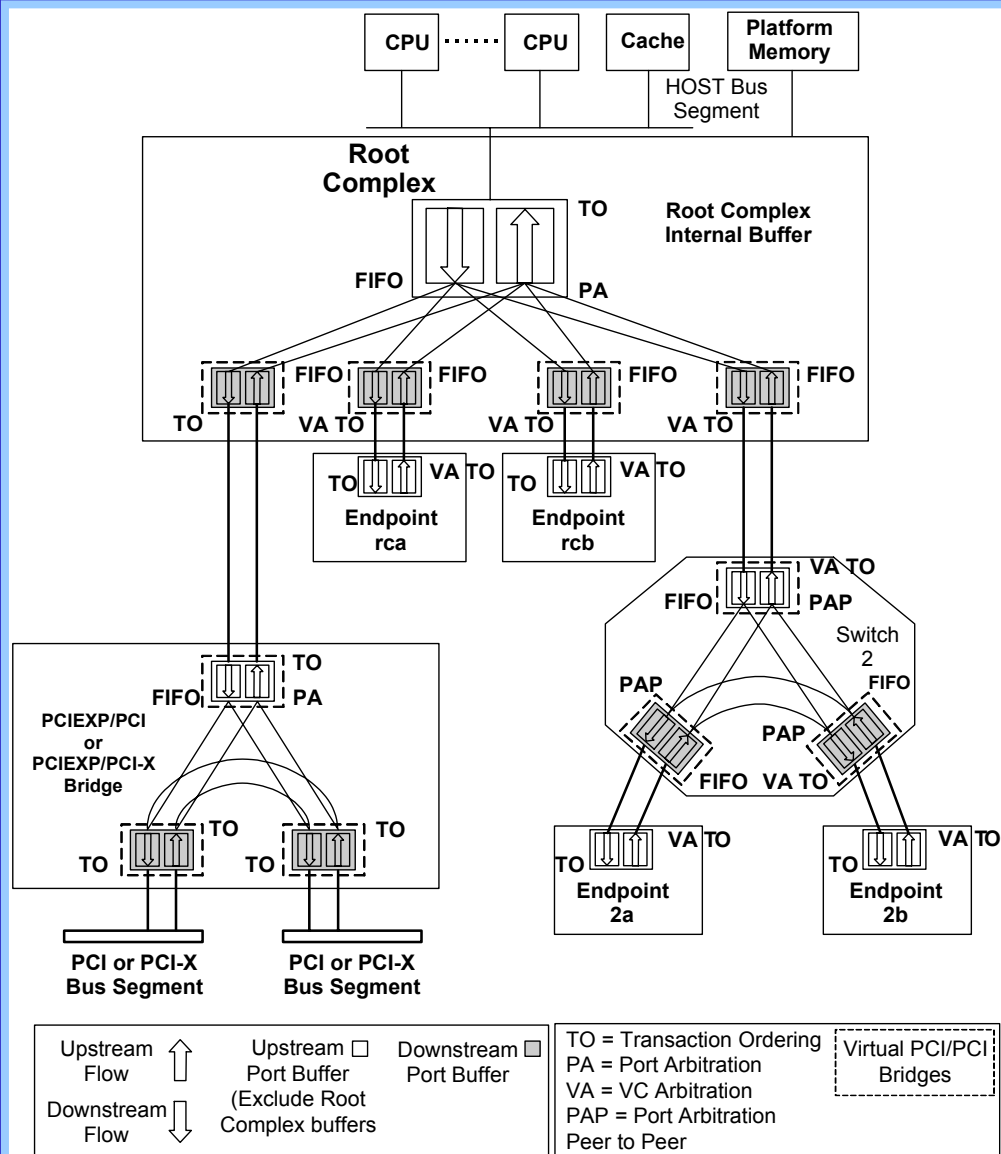
## Flow Control Introduction ... continued

- In order to provide permit software to fine tune the Q of S for different transactions throughout the PCI Express platform the Flow Control protocol is defined:
- Flow Control is discussed in two parts.
  - Part 1: Defines Virtual Channels, Traffic Classes, and determination of available buffer space.
  - Part 2: Defines the Flow Control Protocol for transmitting and receiving TLPs encapsulated in LLTPs.



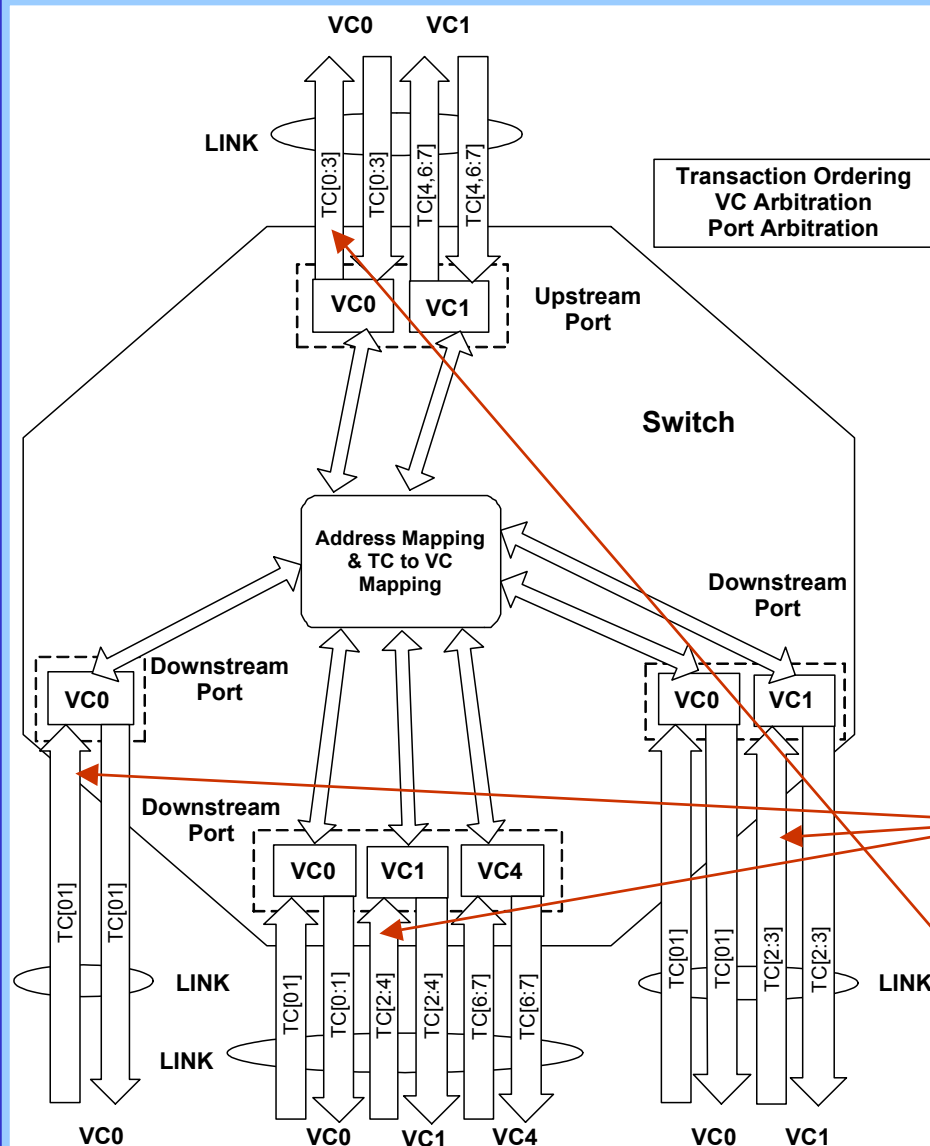
## Flow Control Part 1 ... VC and TC

- The link bandwidth available on each link is based on the physical existence of differentially driven pair of signal lines defined as a lane ( a pair in each direction). A multiple lane link defines multiple pairs of differentially driven signal lines.
- PCI Express defines Virtual Channels (VCs) on each link. The VCs are entirely a virtual concept and NOT a measure of physical differentially driven pair that defines the maximum bandwidth.
- VCs are channels that TLPs can flow through between two PCI Express devices in the virtual sense. It is possible to have a different number of VCs between any two PCI Express devices versus any other two PCI Express devices.
- The software establishes at each transmitting port a number of VCs and a specific set of TLPs can transmit through each VC.



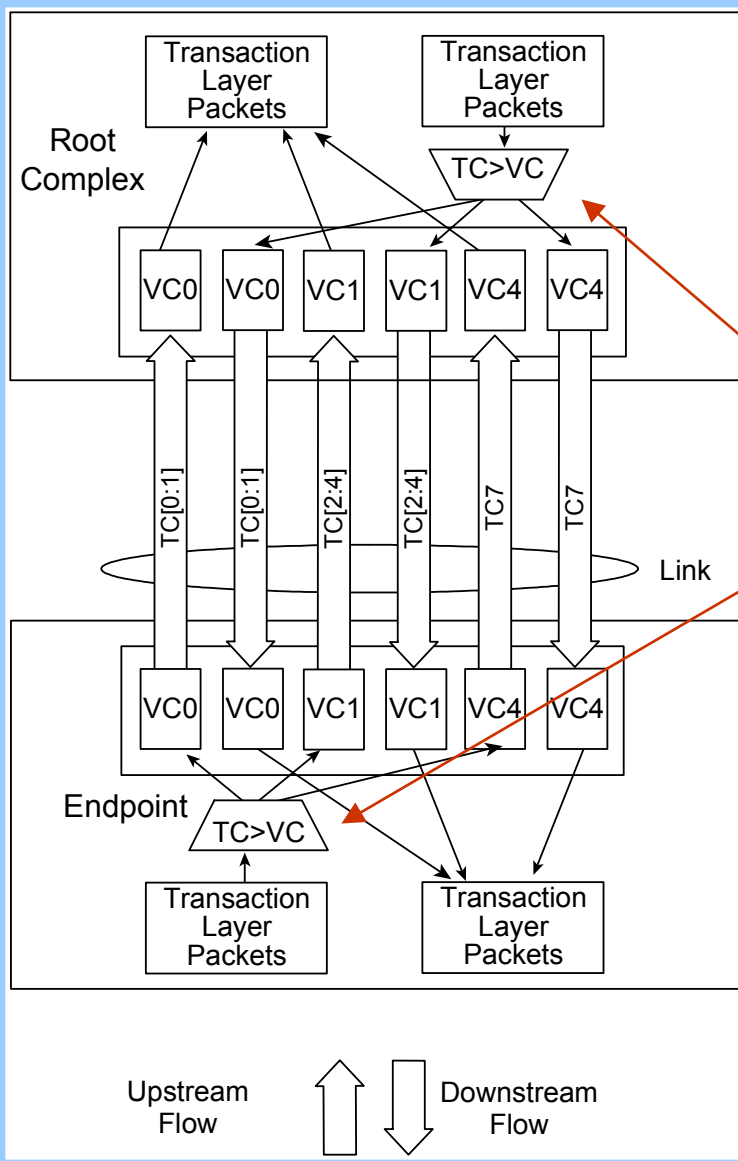
## Flow Control Part 1 ... VC and TC continued

- The set of TLPs that are assigned to each VC is established by the Traffic Class (TC) numbers. The TC number (0 to 7) is assigned to each TLP in the Transaction Layer at the source of the requester transaction. The TLP of the associated completer transaction uses the same TC number.
- Each VC number is assigned a priority for transmission from the buffer onto the link. Consequently, all TLPs assigned to a specific VC number have the same priority as group relative to the TLPs assigned to another VC number. This priority is determined by VC Arbitration.
- Within the each VC number there is a priority among the TLPs assigned to the same VC number via the TC numbers. This priority is determined by Port Arbitration.



## Flow Control Part 1 ... VC and TC continued

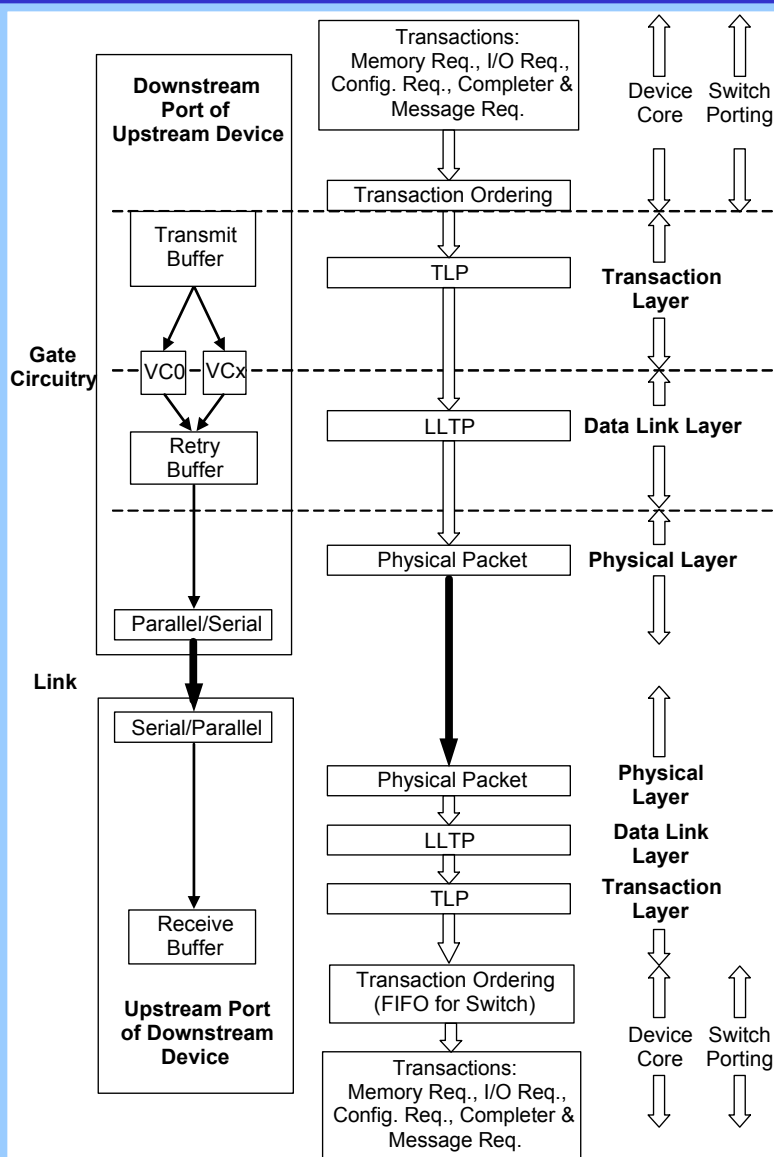
- As previously stated the assignment of a TC number to a TLP occurs at the source of the requester transaction packet.
- Associated with each transmitting port of a PCI Express device software assigns a group of TC numbers to a specific VC number.
- As the TLPs flows throughout the PCI Express platform the TLP is mapped to a specific VC number by the TC number.
- As exemplified in a switch, the TLPs assigned to a specific VC number as they transverse one link may not use the same VC number as it transveres another link.
- For example, upstream flowing TLPs assigned TC 1, 2, and 3 enter the downstream ports on VC 0 , 1, and 1; respectively. The transmission of the TLPs on the upstream port are all on VC0.
- At the upstream port the different VCs are given bandwidth priority by software. The software also maps the TC number to specific VC numbers. In this example TC 1, 2, and 3 are all assigned to VC0.



## Flow Control Part 1 ... VC and TC continued

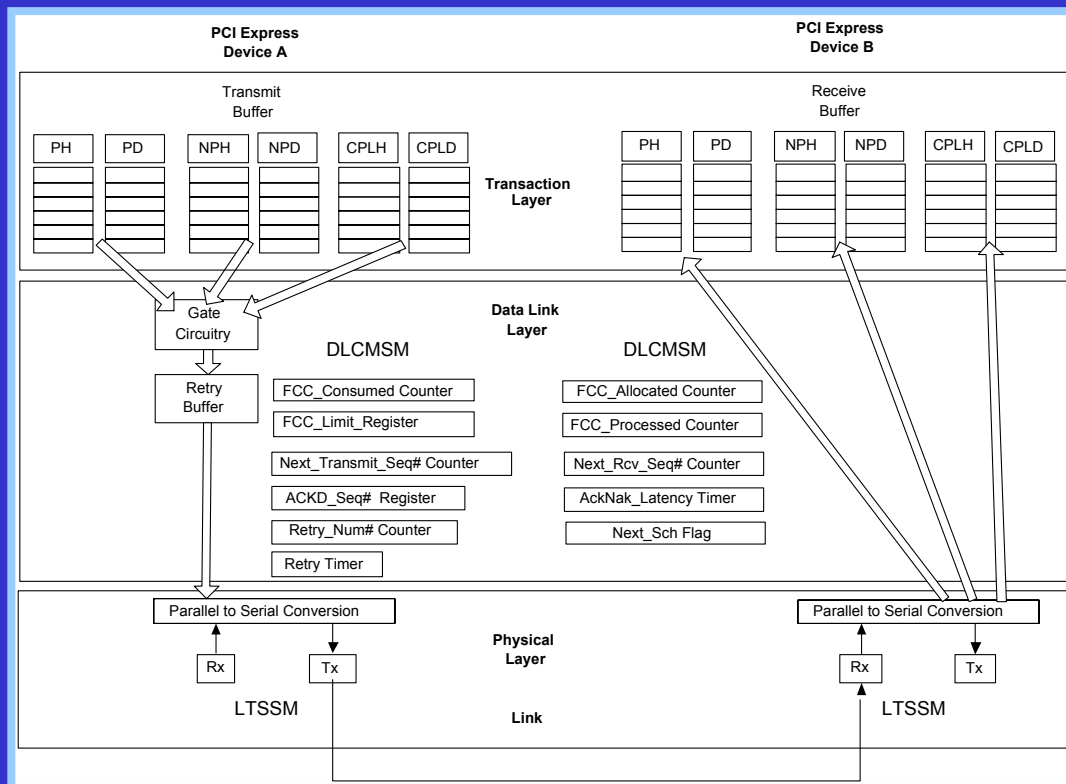
- The mapping for TLPs with specific TC numbers to a specific VC number also occurs on a non-switch PCI Express devices.
- As exemplified in this figure of a link between the Root Complex and an endpoint, the mapping occurs at the respective transmitting ports.
- Also exemplified in this figure is the requirement that TC number mapping to a VC number flowing downstream is symmetric with the upstream mapping.
- Obviously the end point can be replace by a switch and thus it is possible for the switch to re-map the TLP to different VC number on the downstream side of the switch versus the upstream of the switch.
- A bridge can only enable VC0; consequently, all TC numbers must be mapped to VC0.





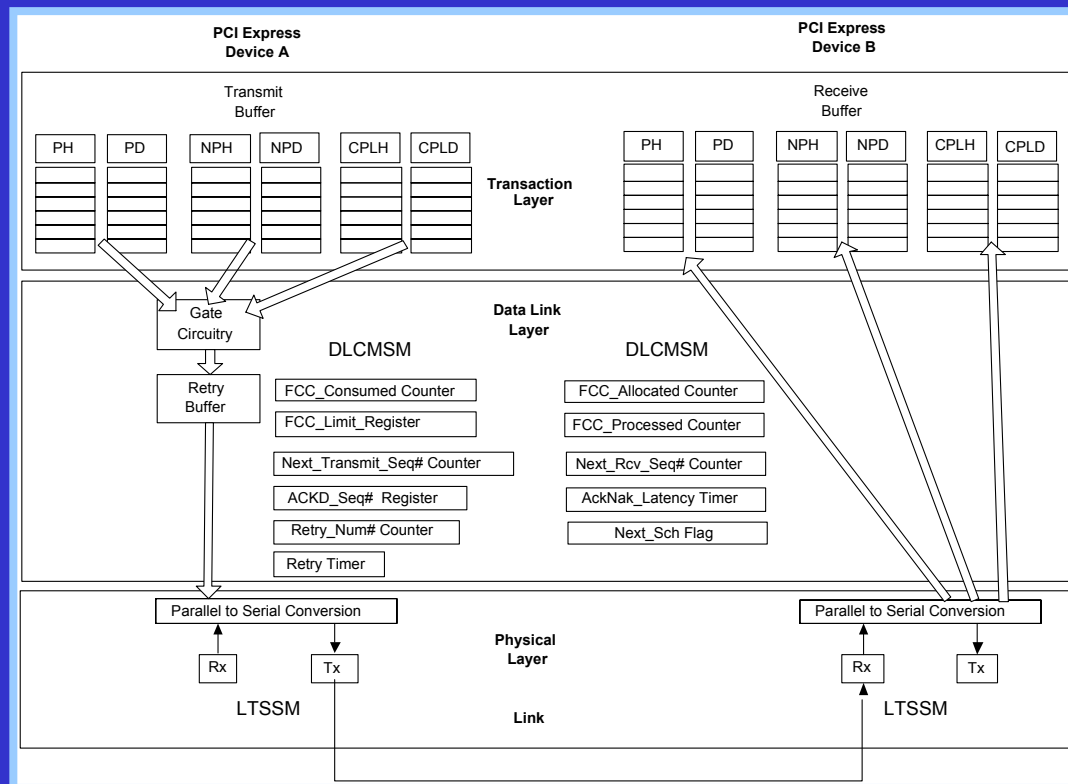
## Flow Control Part 1 ... Determination Available Buffer Space

- Per the previous slides, the transmission of a TLPs across a link is dependent on Transaction Ordering, and the transmission priority of each TC number and VC number. The priority protocols of TLPs transmission due to TC and VC numbers are discussed in later slides.
- One other consideration to the specific TLP transmitted is the size of the TLP.
- The different types of TLPs consist of a Header field, Data field (when applicable) and a Digest field (optional).
- As exemplified in the figure, the TLP will be encapsulated into a LLTP which is contained within a Physical Packet transmitted across the link.
- A TLP is transferred from the transmit buffer as a TLP to the retry buffer as a LLTP per the control of the Gate circuitry.
- The Gate circuitry will only transfer a TLP from the transfer buffer if sufficient buffer space is available at the receiving port (receive buffer).
- Each VC number is assigned a set of buffers at the receiving port. The set of buffers assigned to one VC number is independent of a set of buffers assigned to another VC number.



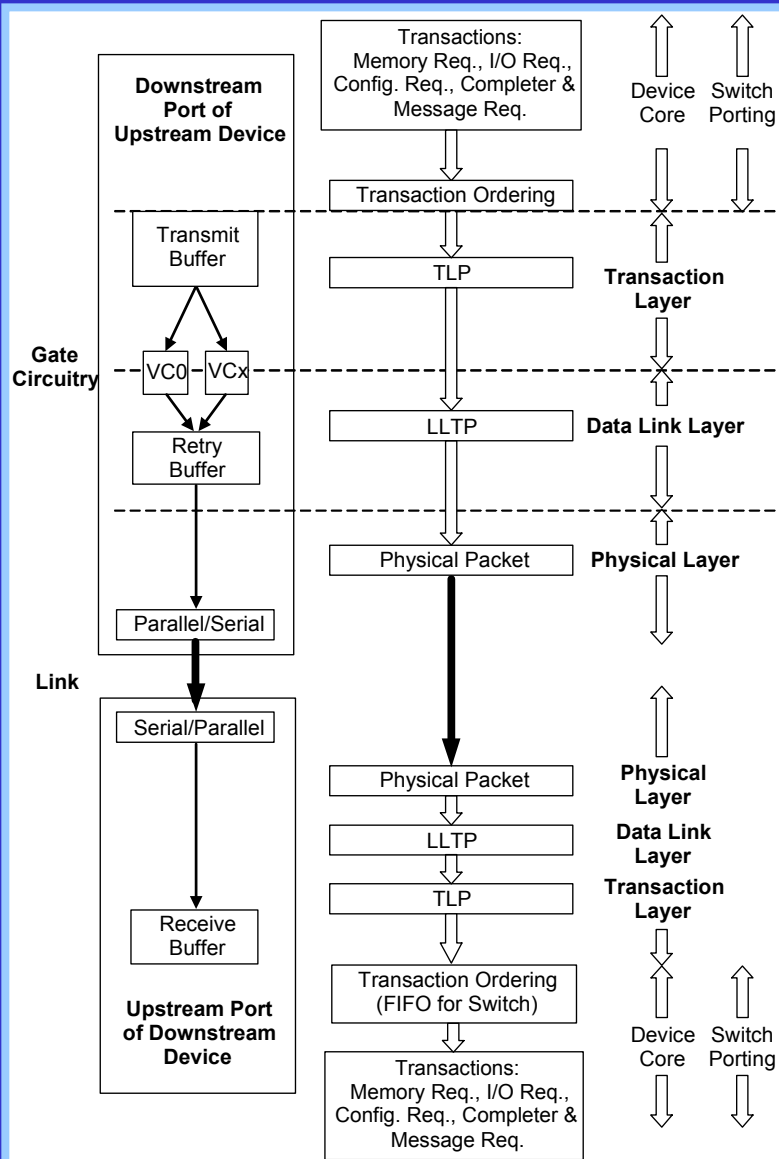
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- Each type of TLP is composed of a Header field, Data field (if applicable) and a Digest field (Optional).
- For each set of buffers assigned to a specific VC number, there are six buffers. The six buffers are defined as follows:
  - PH: The buffer for the Posted Header (memory write for example).
  - PD: The buffer for the Posted Data (memory write for example).
  - NPH: The buffer for Non-Posted Header (I/O read for example).
  - NPD: The buffer for Non-Posted Data (I/O write for example).
  - CPLH: The buffer for completer transaction Header (associated with any requester transaction).
  - CPLD: The buffer for completer transaction with data (associated with I/O read requester transaction for example).



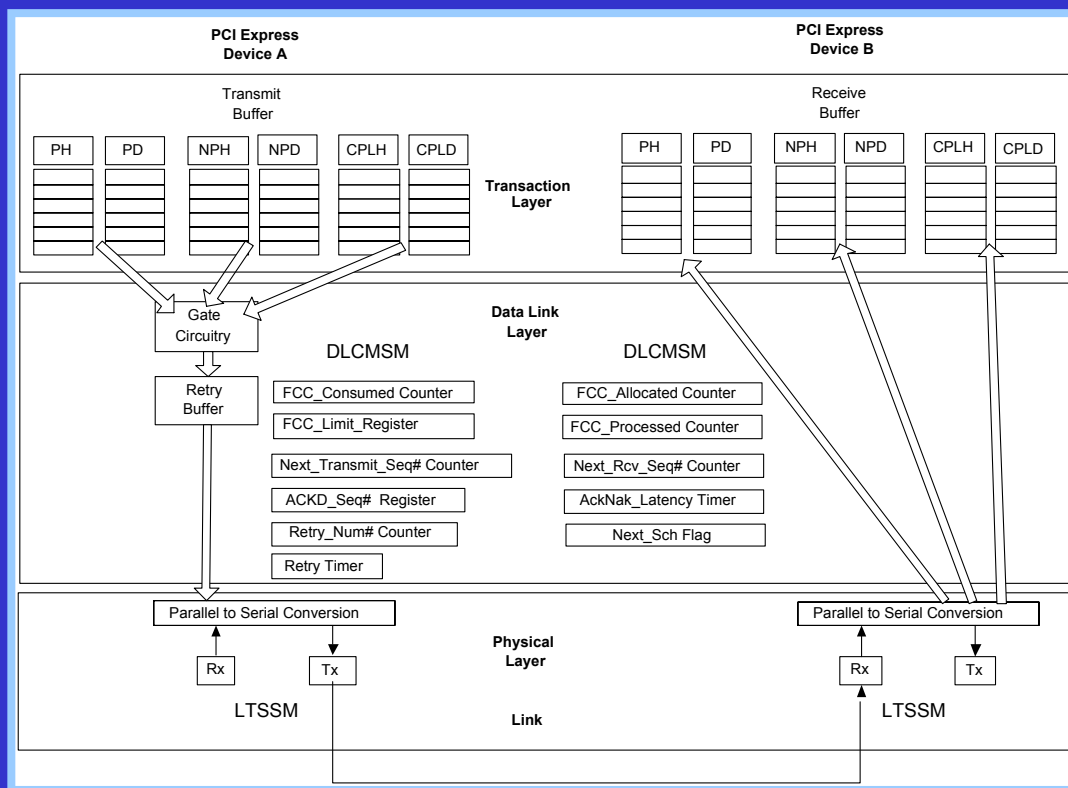
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- Each of the field of the TLP is assigned a value called Flow Control Credit (FCC). Each FCC has a value of 16 bytes.
- The Header field is assigned a value of one
- The Data fields are assigned a value of one FCC for all TLPs except those related to memory address space and message address space.
- The Data fields for TLPs related to the memory and message address spaces are assigned a FCC value equal to the actual Data field size per the LENGTH field divided by four.
- Note: The Digest is optional and thus available buffer space for it is not tracked. It is assumed to be needed and occurs in parallel with the Header fields. The focus for the balance of these slides will not include the Digest field though it is implied.



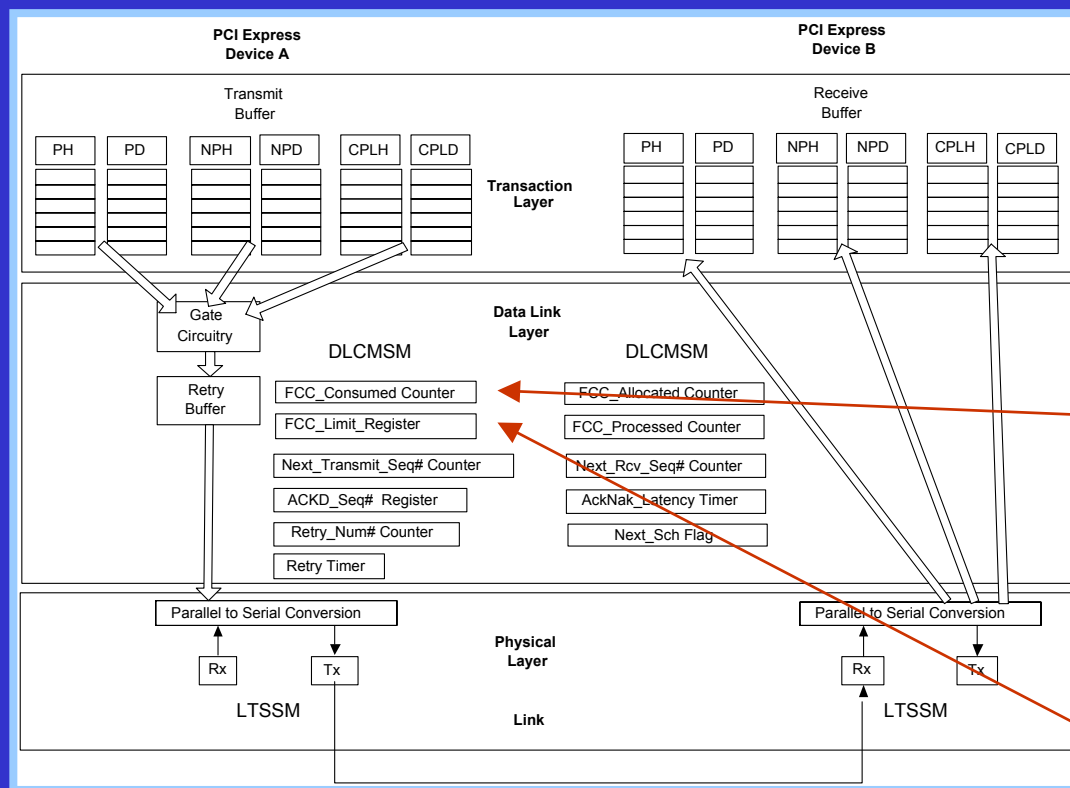
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- The focus of the previous figures has been simplified to the interaction between PCI Express Device A and PCI Express Device B.
- These two PCI Express devices in the following discussion will be viewed as the Root Complex and an endpoint.
- It is also possible that one of the PCI Express devices under consideration a switch.
- The figure provides the visual interpretation for a switch. The concepts of the transmit and receiver buffers, Gate circuitry, and retry buffer are the same.
- The only difference is that the PCI Express device core is replaced by a switch porting mechanism



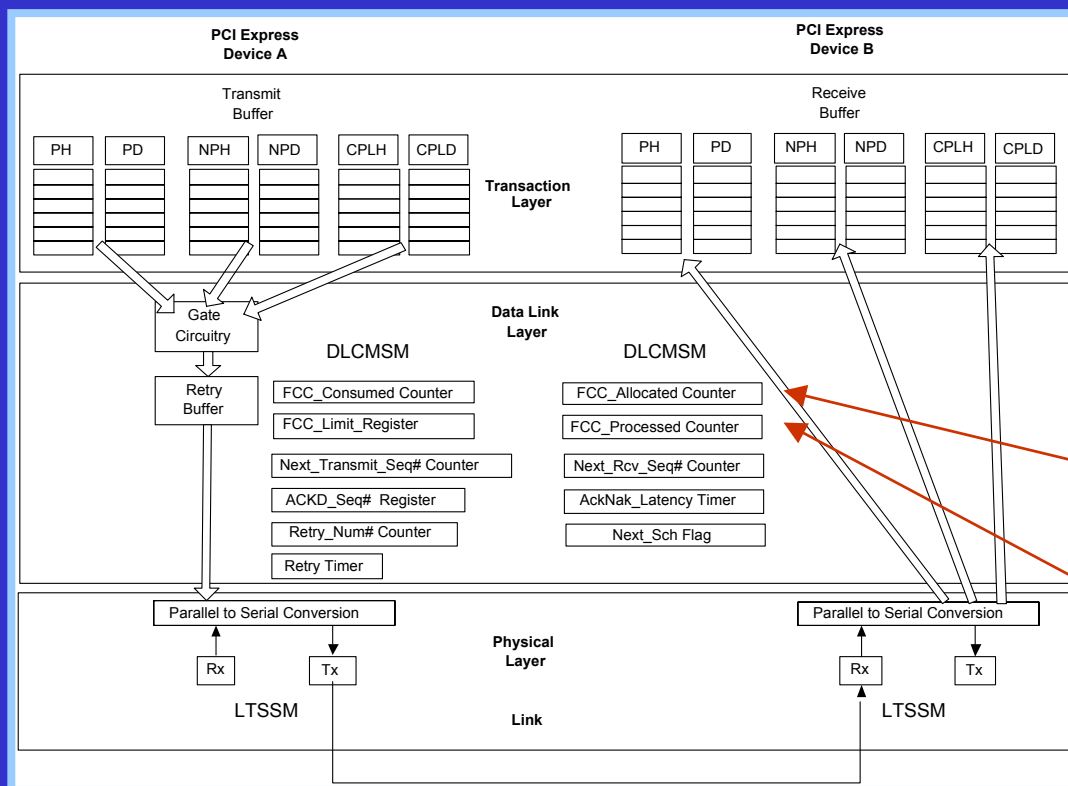
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- Returning to the basic figure, to simplify the discussion assume that the PCI Express are the Root Complex (Device A) and endpoint (Device B) connected by a single link without intervening switches.
- Assume that Device A is transmitting TLPs to Device B and thus **Device A is the transmitting port** and **Device B is the receiving port.**



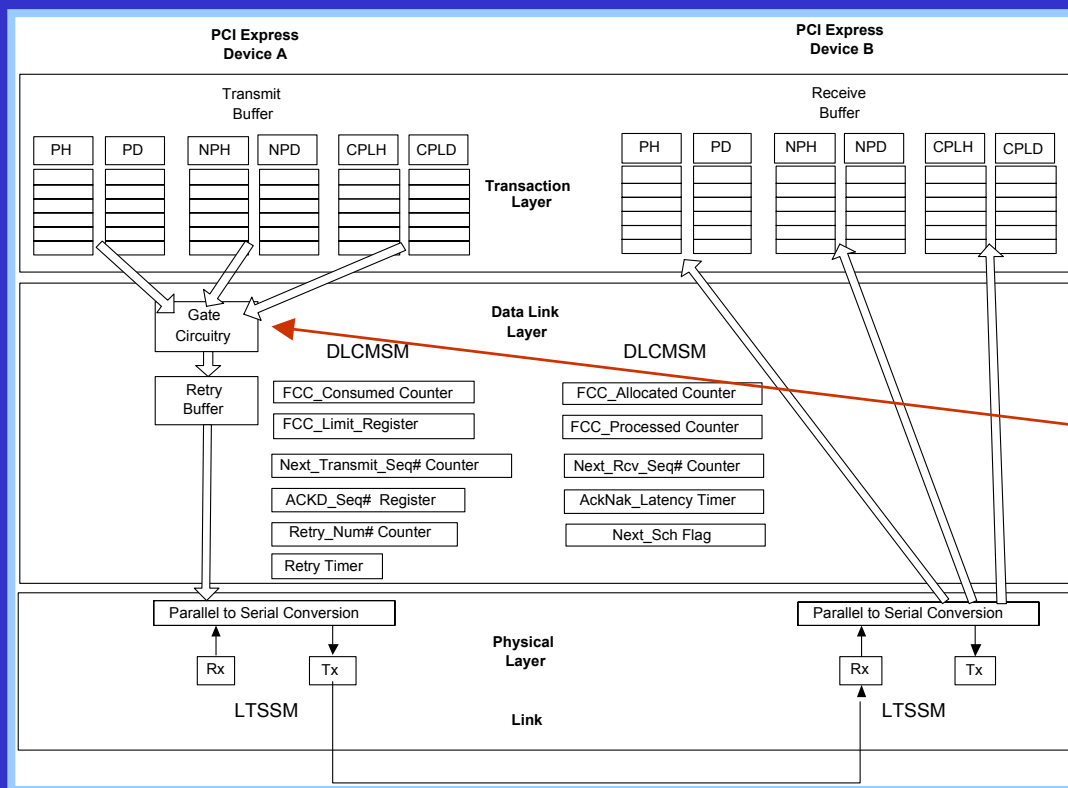
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- In order for the transmitting port to know if there is available buffer space at the receiving port the following entities are defined for each of the buffers in the buffer set of a specific VC number::
  - FCC\_Consumed Counters: These contain the total FCC value of all TLPs transferred to the retry buffer. Once a TLP is transferred to the retry buffer (encapsulated into the LLTP) the transmitting port assumes that the TLP will be transmitted to the receiving port
  - FCC\_Limit Registers: These contain the FCC value of the available buffer space at the receiving port.



## Flow Control Part 1 ... Determination Available Buffer Space ... continued

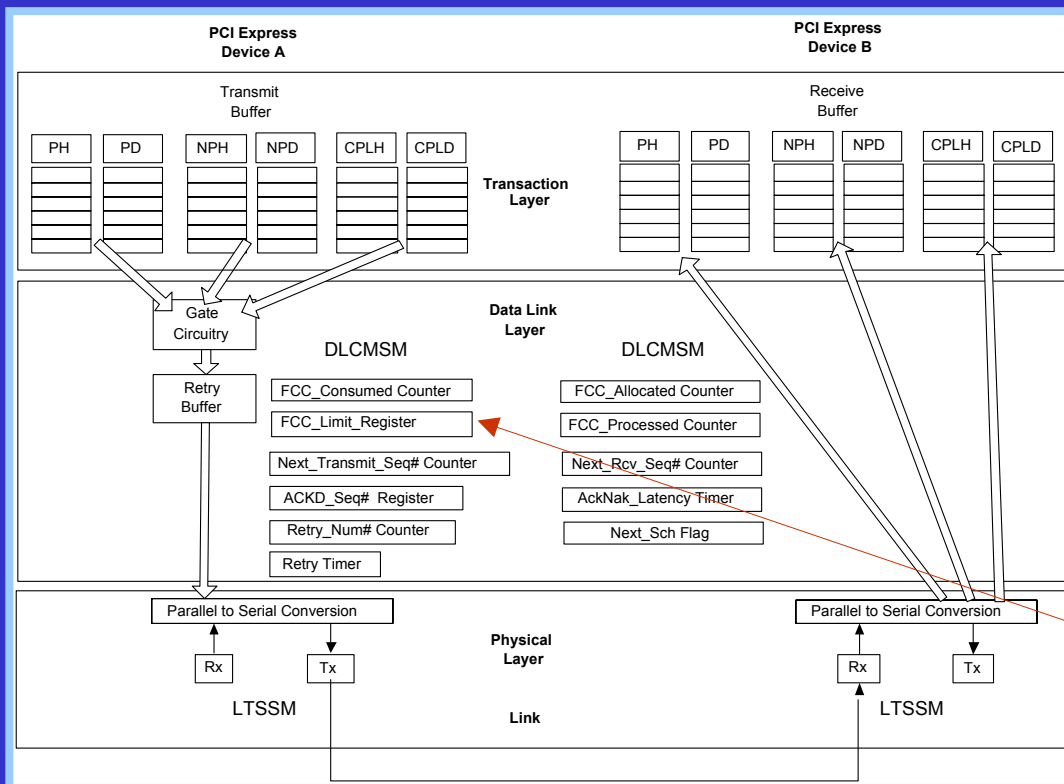
- In order to keep track of available buffer space at the receiving port the following entities are defined for each of the buffers in the buffer set of a specific VC number::
  - **FCC \_Allocated Counters:** These contain the total FCC value allocated since Flow Control Initialization had occurred.
  - **FCC\_Processed Counters:** These contain the FCC value of the TLPs received since the update of the FCC\_Allocated Counter.



## Flow Control Part 1 ... Determination Available Buffer Space ... continued

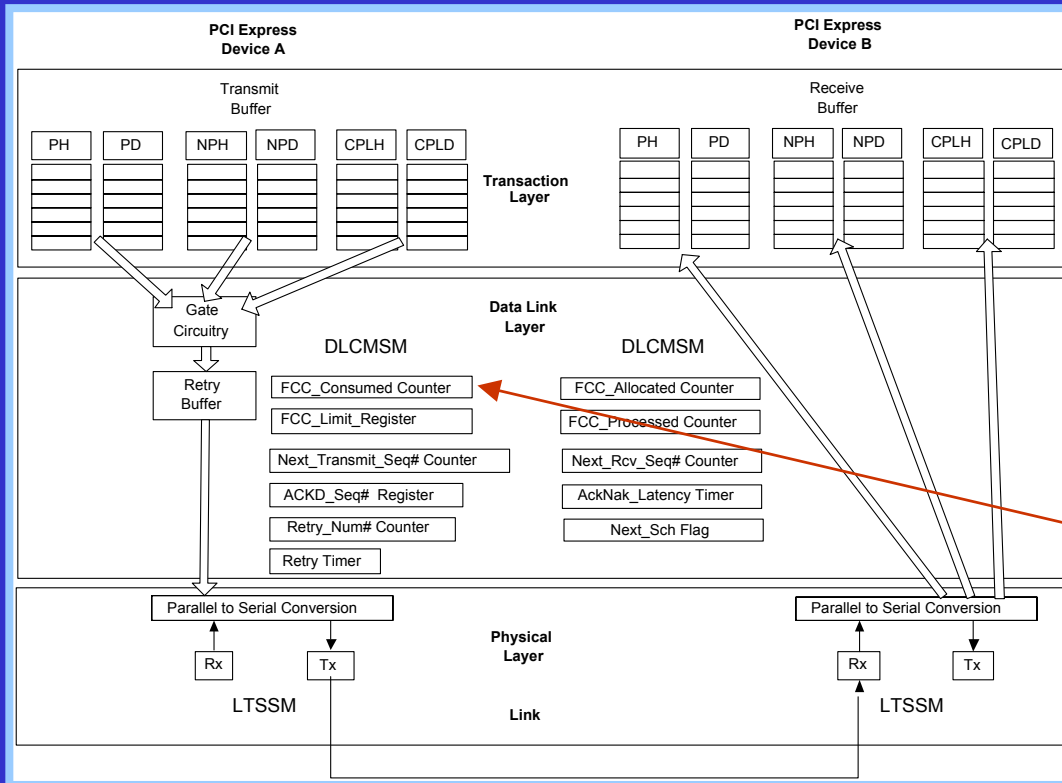
- Whenever the Gate circuitry in the transmitting port wants to transfer a TLP from the Transaction Layer, it must first determine if buffer space is available at the receiving port. It determines the FCC value of the TLP under consideration and defines it as the FCC\_Pending Value.





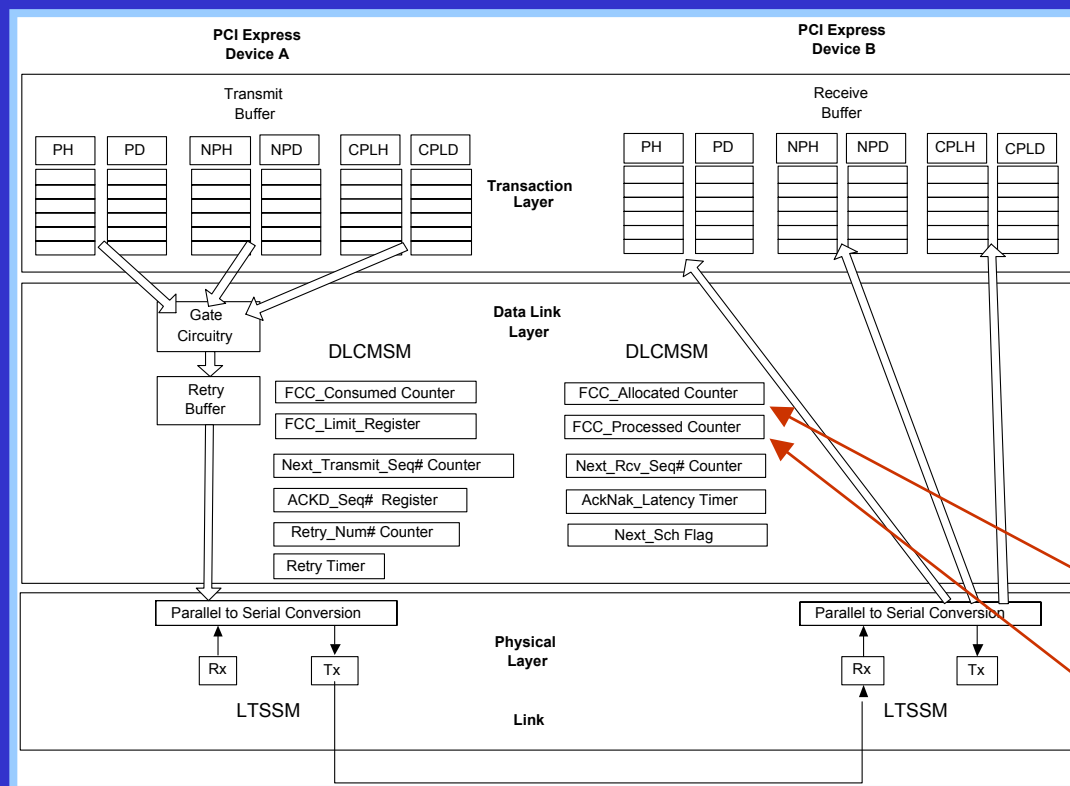
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- The protocol for transmitting a TLP encapsulated in a LLTP contained within a Physical Packet for a specific VC number is as follows:
  - Upon entry into the L0 link state the Flow Control Initialization protocol establishes the available buffer space for each of the six buffers in the receiving port via InitFC1 and InitFC2 DLLPs.
  - The initial available buffer space for each of the six buffers is contained in these DLLPs and is stored into the FCC\_Limit Register.



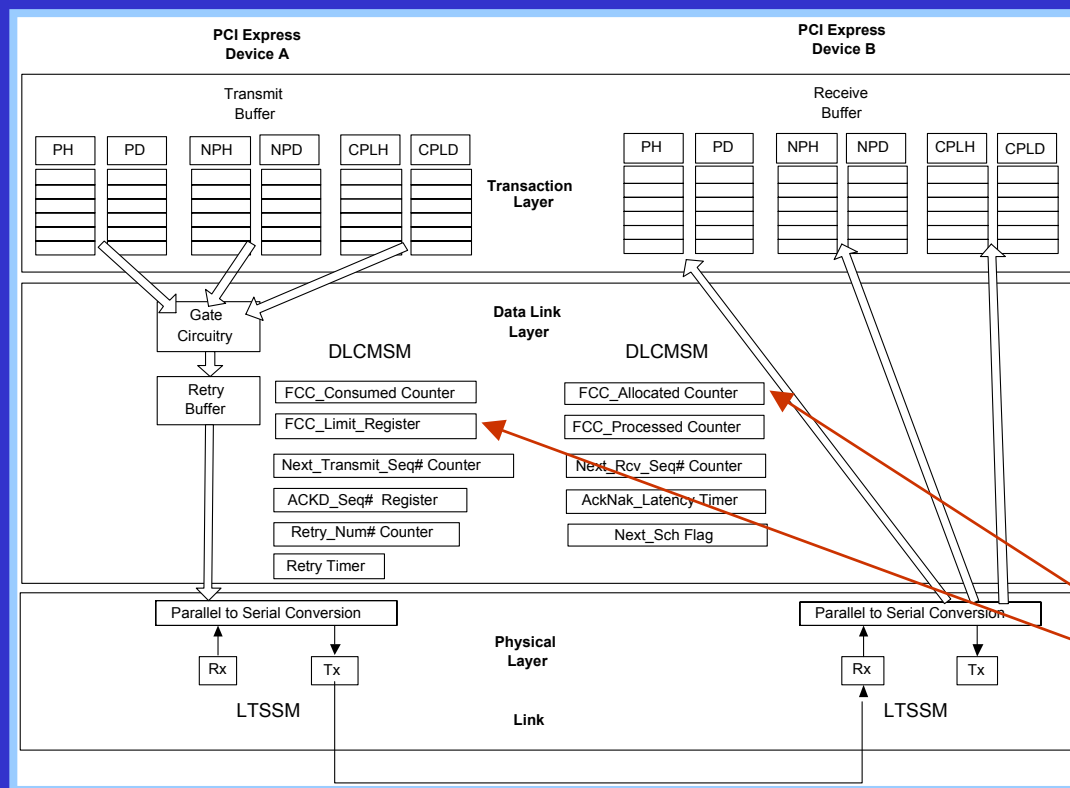
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- The protocol for transmitting a TLP encapsulated in a LLTP contained within a Physical Packet for a specific VC number is as follows: ... continued
  - The FCC\_Pending Value is determined by the Gate Circuitry and matched against the difference between the values in the FCC\_Limit Register and the FCC\_Consumed Counter.
  - If sufficient space is available the TLP is transferred to the Data Link Layer and the FCC\_Pending Value is added to the FCC\_Consumed Counter.
  - The mathematical protocol for comparing the difference between the values in the FCC\_Limit Register and the FCC\_Consumed Counter is via transfer equations detailed in the Book.



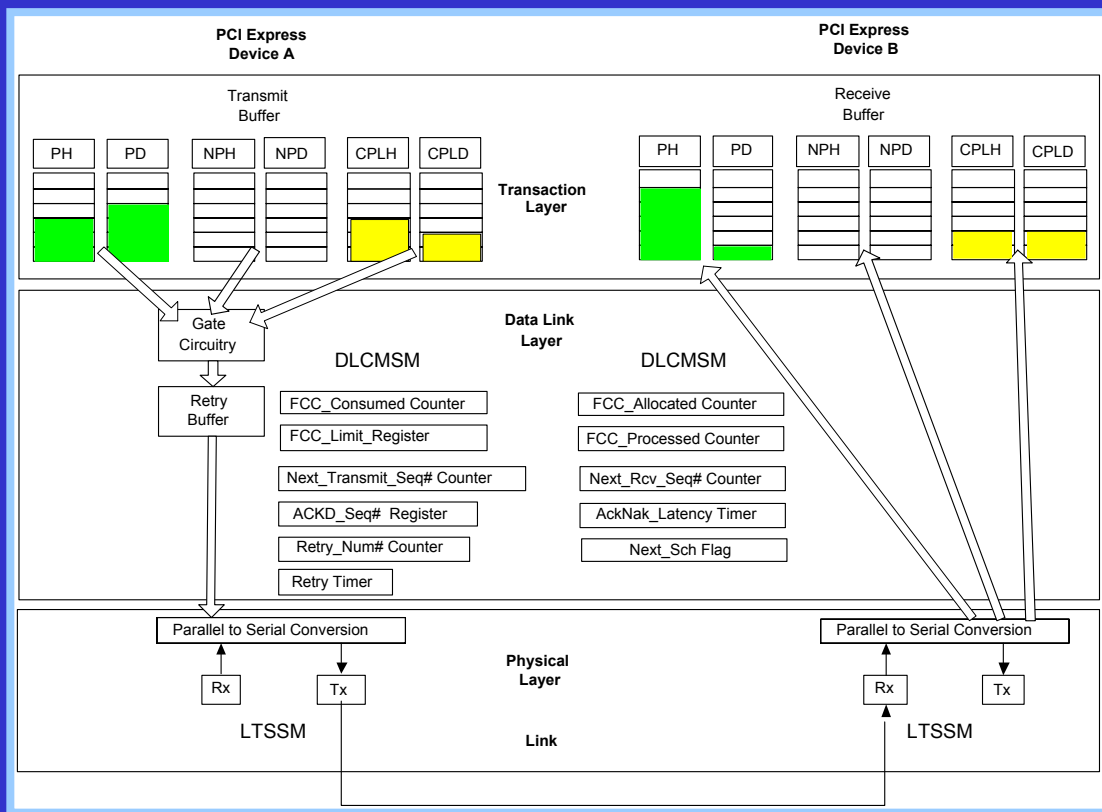
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- The protocol for receiving a TLP encapsulated in a LLTP contained within a Physical Packet for a specific VC is as follows:
  - Upon entry into the L0 link state the Flow Control Initialization protocol has established the available buffer space for each of the six buffers in the receiving port via InitFC1 and InitFC2 DLLPs.
  - The initial available buffer space for each of the six buffers is contained in these DLLPs and is stored into the FCC\_Allocated Counters.
  - The FCC value of the TLP received is added to the FCC\_Processed Counter.



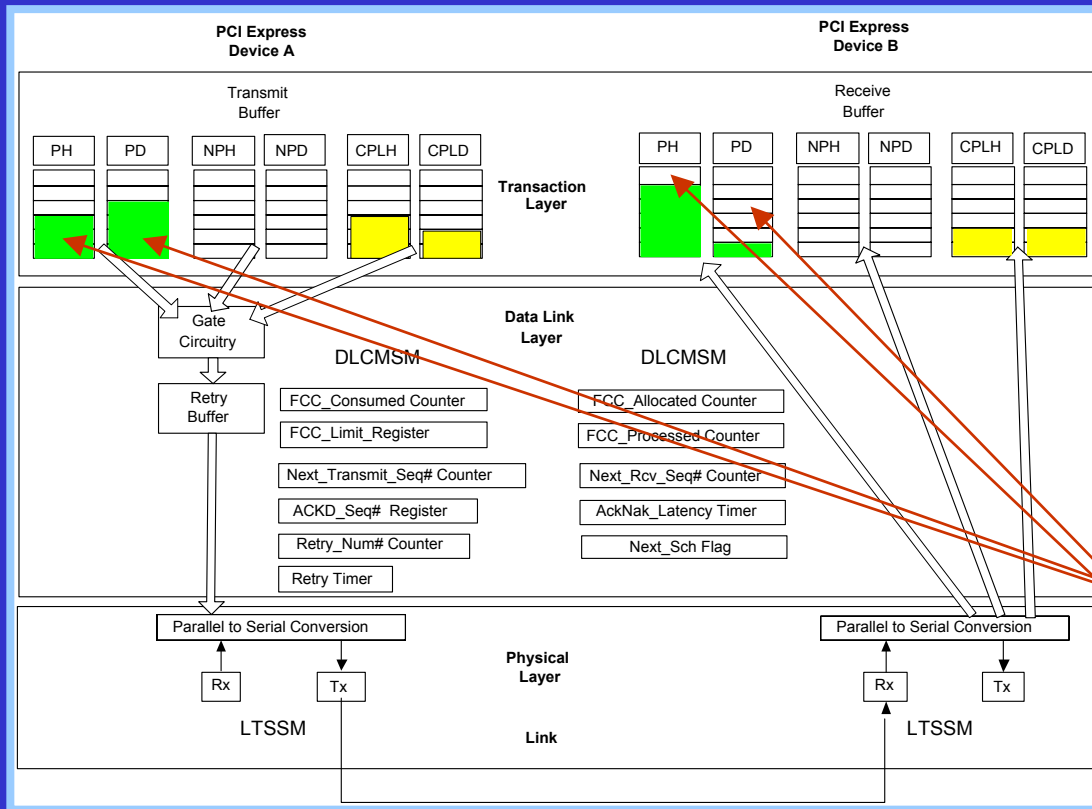
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- The protocol for receiving a TLP encapsulated in a LLTP contained within a Physical Packet for a specific VC is as follows: ... continued
  - When the TLP is removed from the receive buffer and processed in other parts of the Transaction Layer the value in the FCC\_Processed Counter is added to the value in the FCC\_Allocated Counters.
  - Per the Flow Control protocol the value of the FCC\_Allocated Counter is transmitted to the FCC\_Limit Register in the transmitting port via the UpdateFC DLLPs.



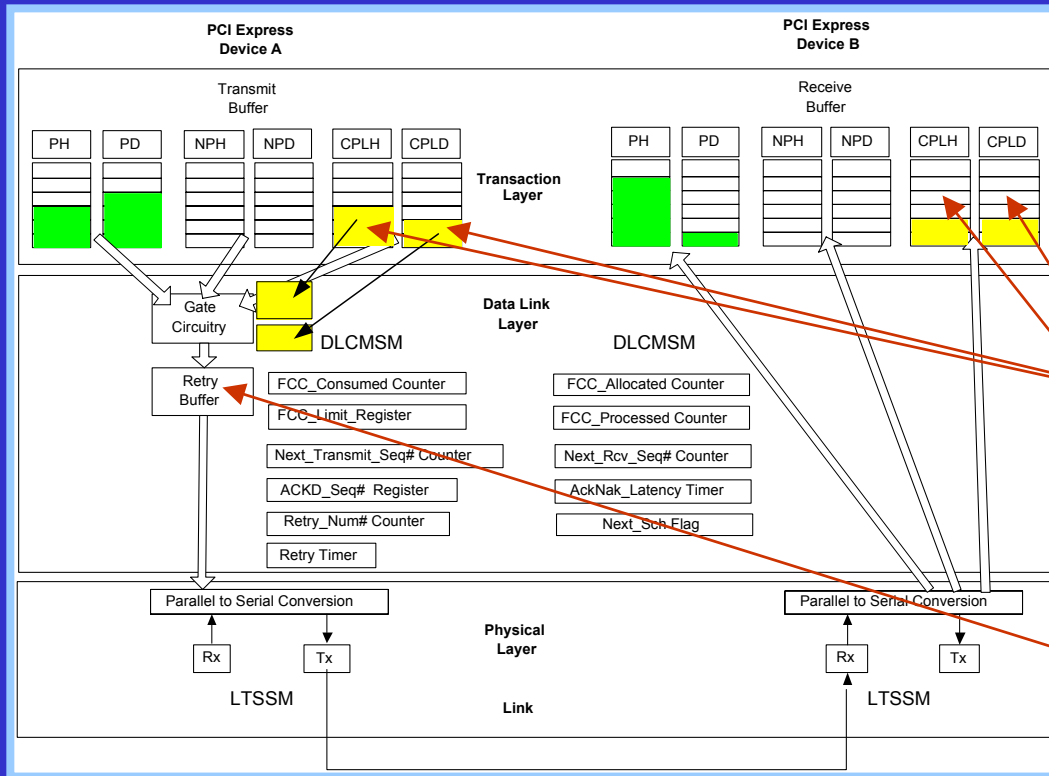
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- There are several items that should be pointed out relative to determining if sufficient buffer space is available at the receiving port.
  - As discussed in the Book, the transfer equations rely on one's complement arithmetic, comparison to boundary conditions of 1/2 the maximum size of the various counters, and the use of 00h and 000h for infinite values. As certain counters rollover the comparison of counters to registers to determine the available buffer space works. See the Book for a detailed example.



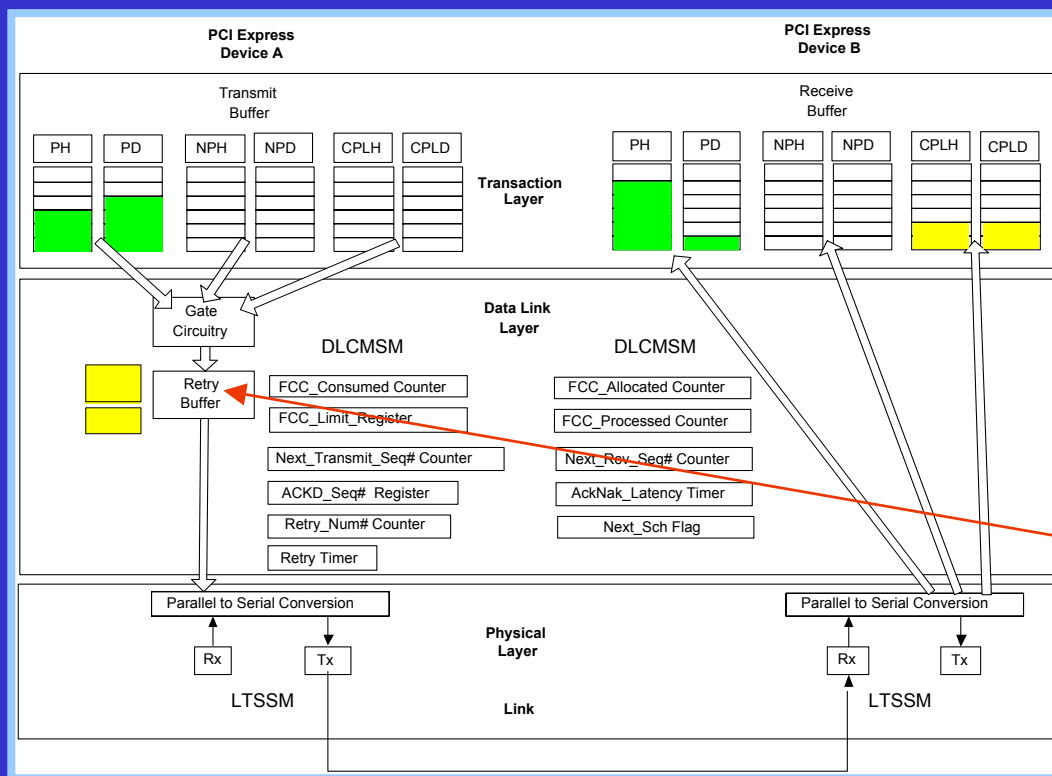
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- There are several items that should be pointed out relative to determining if sufficient buffer space is available at the receiving port ... continued
  - When the Gate circuitry considers a TLP for transmission it must check the available buffer space for both the Header field and Data field (if applicable)
  - For example, a TLP with a posted requester transition is considered (Green). The Gate circuitry determines that the PD buffer at the receiving port had sufficient space available but the PH does not. It must not transfer the TLP to the Data link layer.



## Flow Control Part 1 ... Determination Available Buffer Space ... continued

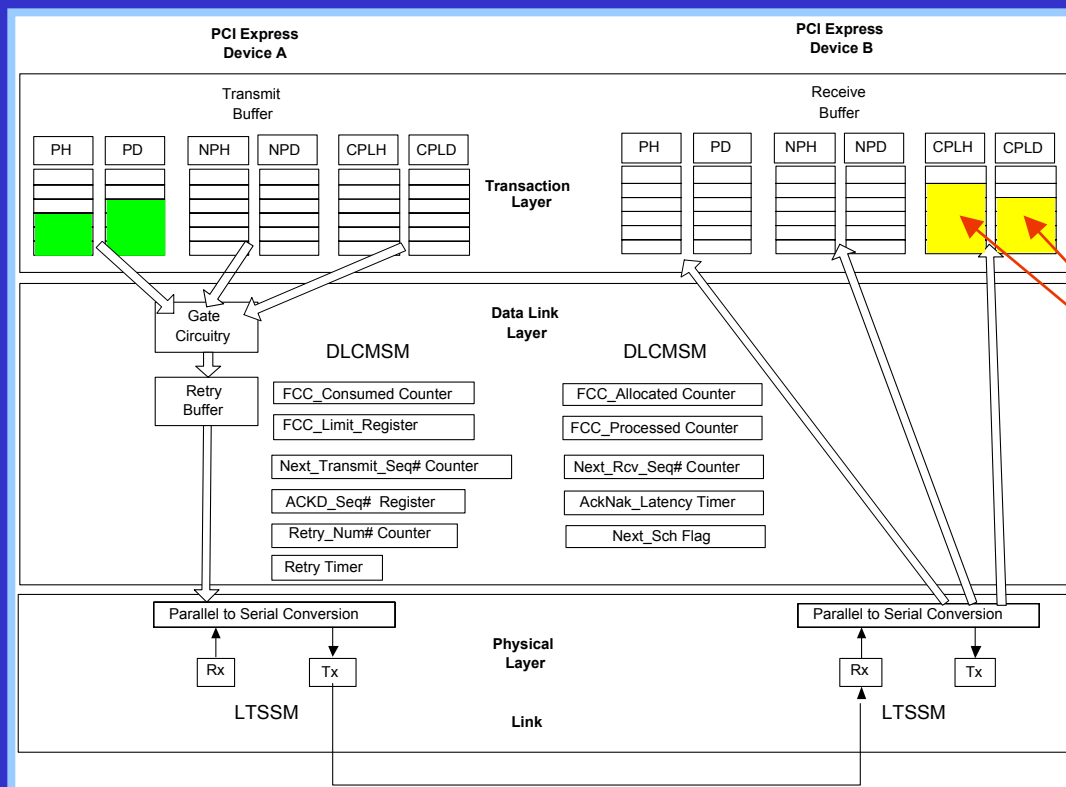
- There are several items that should be pointed out relative to determining if sufficient buffer space is available at the receiving port .. continued
  - For example, a TLP with a complete transition is considered (Yellow). The Gate circuitry determines that the CPLH and CPLD buffers at the receiving port has sufficient space available and thus the Gate circuitry transfers the TLP under consideration to the retry buffer.



## Flow Control Part 1 ... Determination Available Buffer Space ... continued

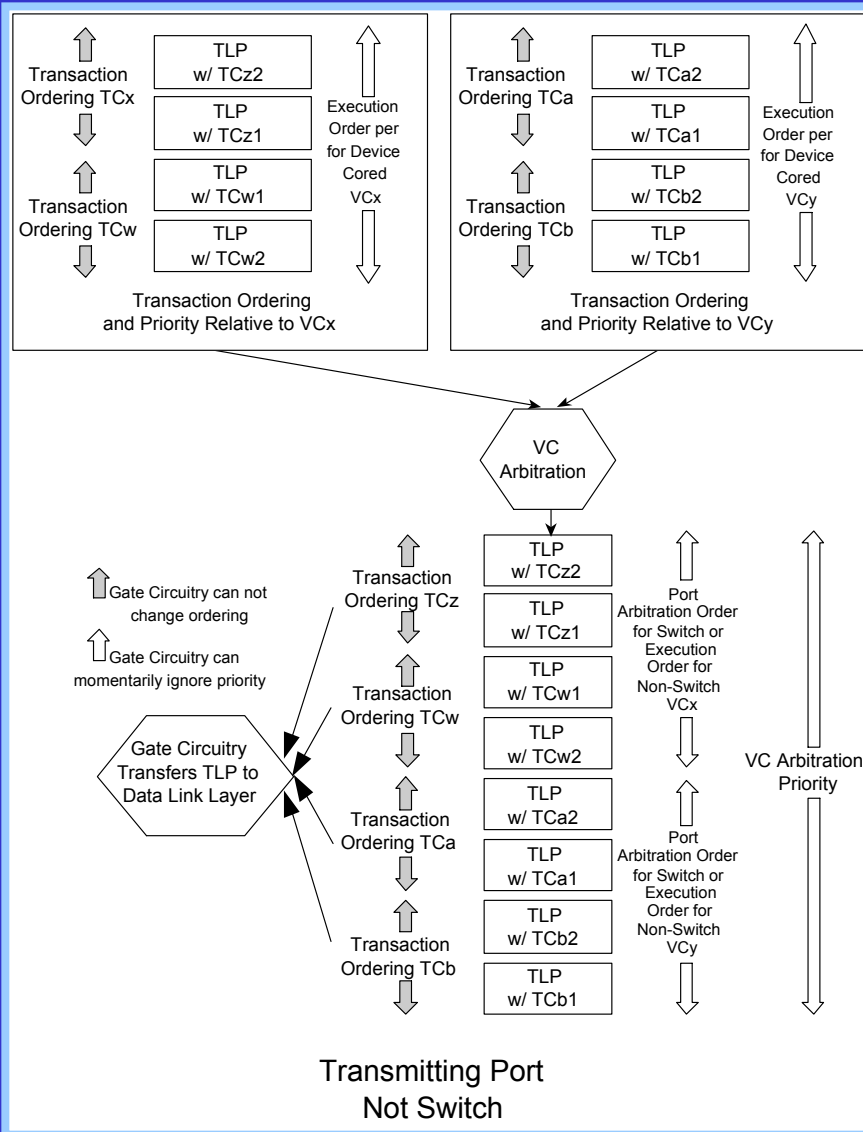
- There are several items that should be pointed out relative to determining if sufficient buffer space is available at the receiving port .. continued
  - The CPLH and CPLD are placed into retry buffer and transmission is attempted multiple times until successfully received at the receiving port.





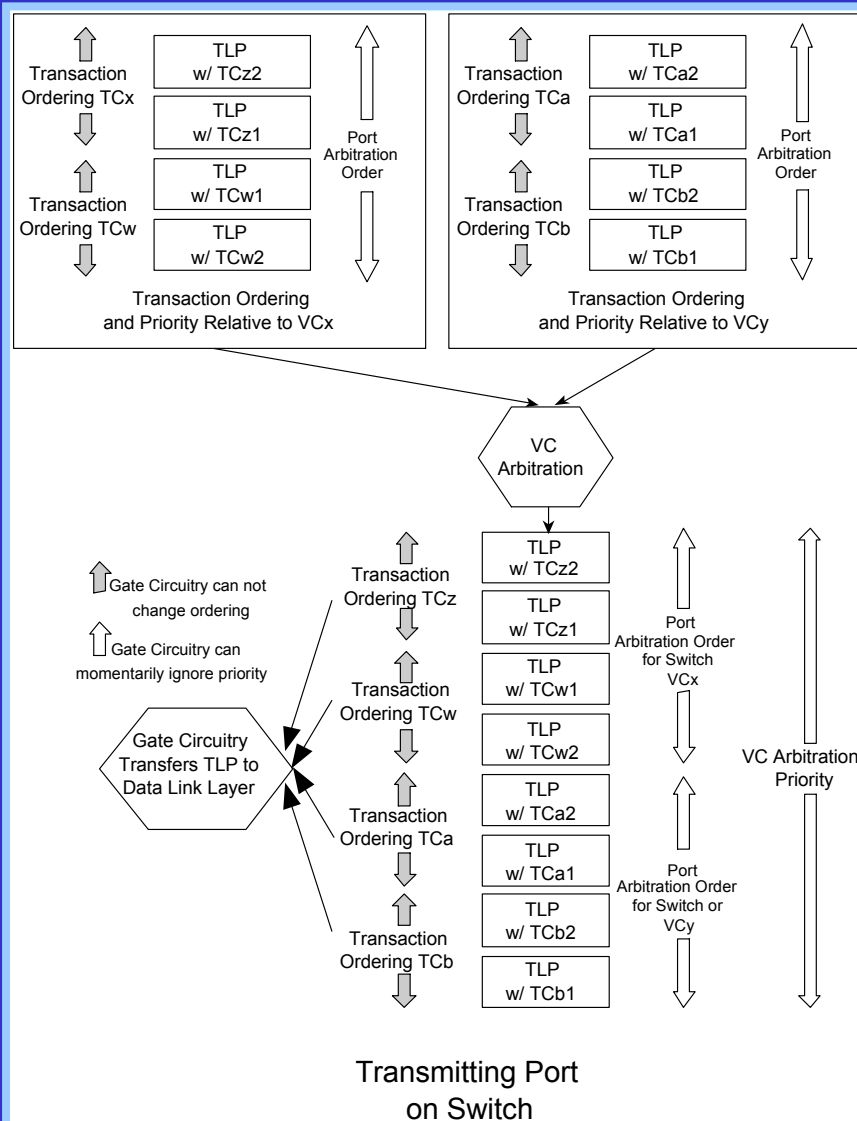
## Flow Control Part 1 ... Determination Available Buffer Space ... continued

- There are several items that should be pointed out relative to determining if sufficient buffer space is available at the receiving port .. continued
  - Subsequently the contents of the retry buffer are transmitted to the receiving port and stored into the receive buffer (Yellow) .
  - Simultaneously and independent of the TLP for the completer transaction just received, a TLP for the posted requester transition is processed and removed from the receive buffer.



## Flow Control Part 1 ... Transmission Summary

- In summary of the Flow Control Part 1 discussion, the transmission of TLPs from **a PCI Express device core** can not be done without consideration of the transmission order.
- Transaction Ordering must be considered for livelock and deadlock considerations.
- The transmission order is determined between TLPs mapped to the same VC number via the mapping of the assigned TC number.
- The priority of transmission between the groups of TLPs mapped to a specification VC number is via VC Arbitration.
- The Gate circuitry must select a TLP to transfer to the Data Link Layer. The Gate circuitry is restricted to the TLPs that will "fit" into receiving port's buffers and that the selection will not violate the aforementioned transmission order.

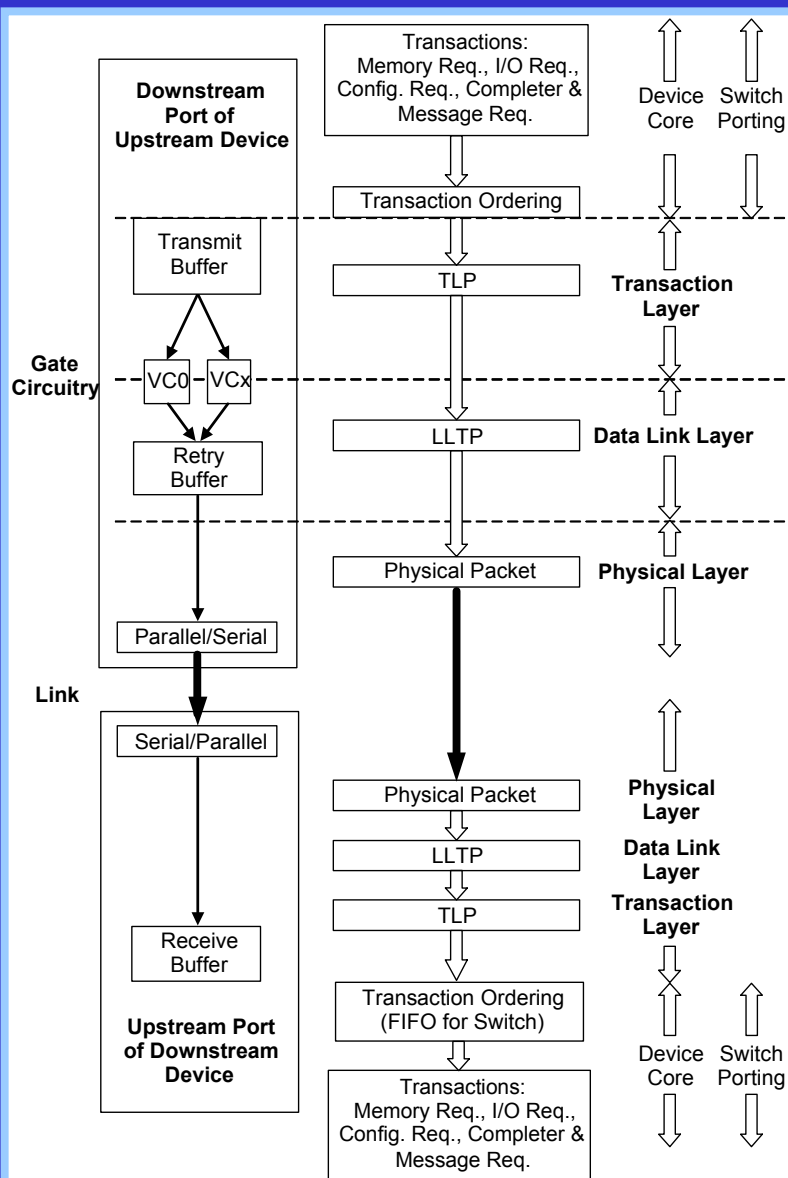


## Flow Control Part 1 ... Transmission Summary ... continued

- In summary of the Flow Control Part 1 discussion, the transmission of TLPs ported through a switch can not be done without any consideration of the transmission order.
- Transaction ordering must be considered for livelock and deadlock considerations.
- The priority of transmission is dependent on the receiving port that is providing the TLP via the Port Arbitration protocol.
- The transmission order is determined between TLPs mapped to the same VC number via the mapping of the assigned TC number.
- The priority of transmission between the groups of TLPs mapped to a specification VC number is via VC Arbitration.
- The Gate circuitry must select a TLP to transfer to the Data Link Layer. The Gate circuitry is restricted to the TLPs that will "fit" into receiving port's buffers and that the selection will not violate the aforementioned transmission order.
- The Flow Control Part 2 will review the priority schemes of the VC and Port Arbitration.

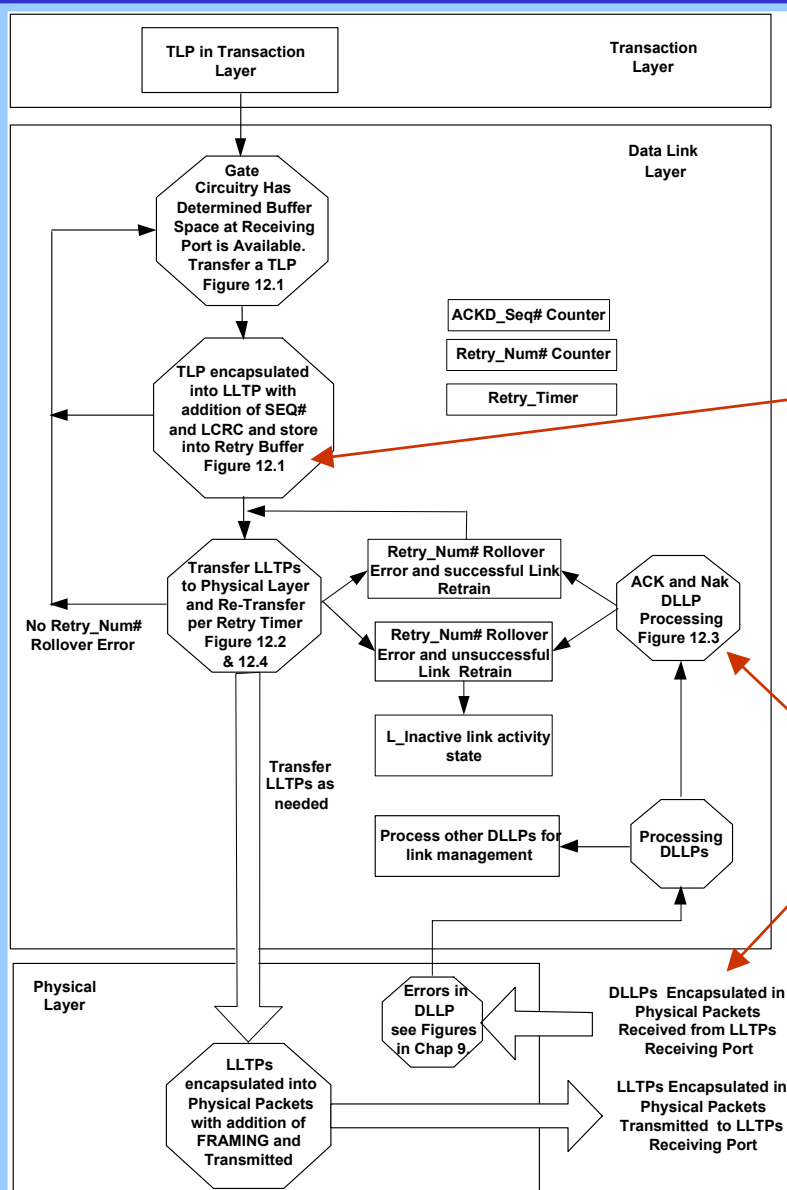
# Chapter 12

## Flow Control Protocol Part 2



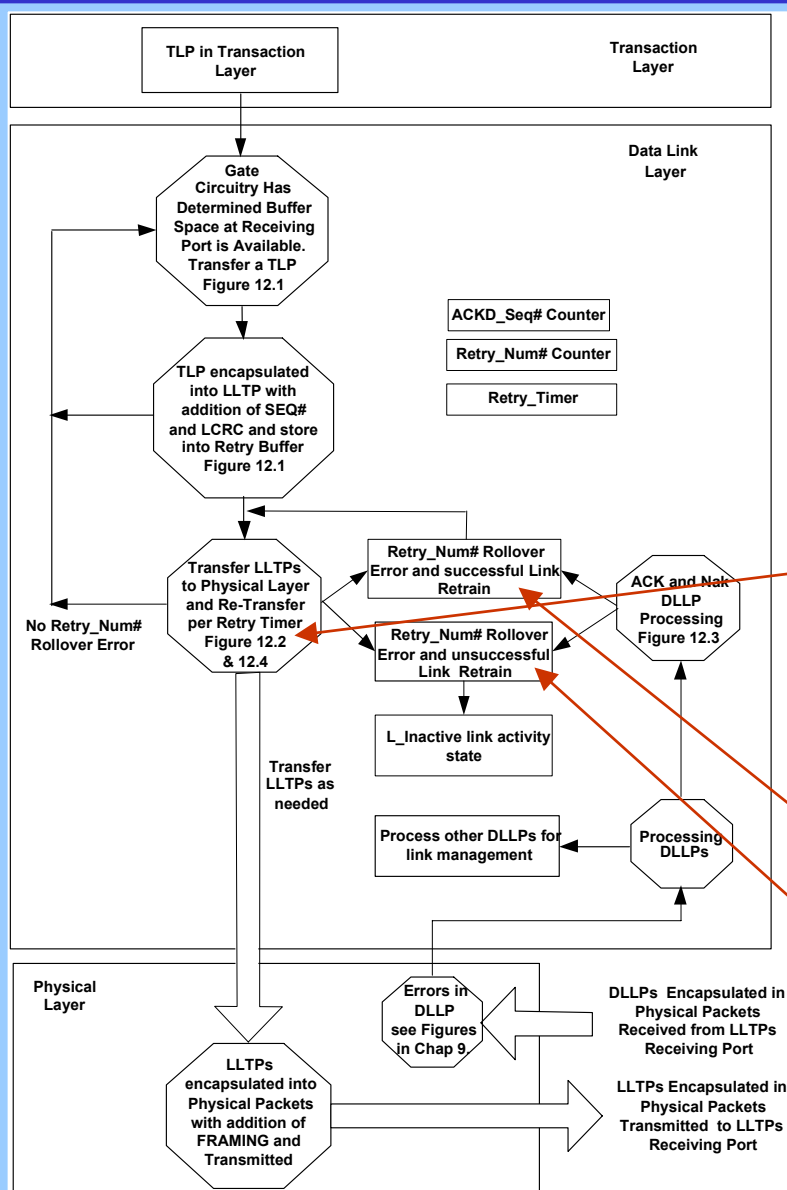
## Flow Control Part 2 ... Protocol for Transmitting and Receiving LLTPs

- Transmitting LLTPs ... Introduction
  - The previous slides discussed how the Gate circuitry in the TLPs' transmitting port must select the next TLP to be transmitted when sufficient buffer space is available at the TLPs' receiving port.
  - Once the Gate circuitry transfers a TLP to the Data Link Layer, it is encapsulated into a LLTP with addition of the SEQ# and the LCRC. The next SEQ# to be applied is stored in the ACKD\_Seq# Counter.
  - The LLTP is placed in the retry buffer in the Data Link Layer. As discussed below the LLTP is subsequently copied and transmitted onto the link within a Physical Packet.
  - The focus of this and subsequent slides is the transmitting of this and other LLTPs stored in the retry buffer.



## Flow Control Part 2 ... Protocol for Transmitting and Receiving LLTPs ... continued

- Transmitting LLTPs Flow Control protocol ... **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - The Gate circuitry in the Data Link Layer will continue to transfer and store TLPs from the Transaction Layer into the retry buffer as LLTPs.
  - As each LLTP is placed into the retry buffer a copy is transmitted across the link via a Physical Packet.
  - The assumption by the transmitting port is one of two events will occur: acknowledgement by LLTPs' receiving port or Retry\_Timer timeout.
  - The LLTPs' receiving port will acknowledge to the LLTPs' transmitting port the receipt of the LLTPs. Acknowledgment via the Ack or Nak DLLPs. Multiple Ack and Naks DLLPs are received from the receiving port as it processes LLTPs.
    - The Ack or Nak DLLPs indicates the SEQ# of the last successfully received LLTP. All of the LLTPs in the retry buffer with a SEQ# equal to or earlier than that received in the Ack or Nak DLLP will be defined as acknowledged and removed from the retry buffer.
  - As discussed in earlier slides the Physical Packets and DLLPs must also be received error free to be processed as discussed above.



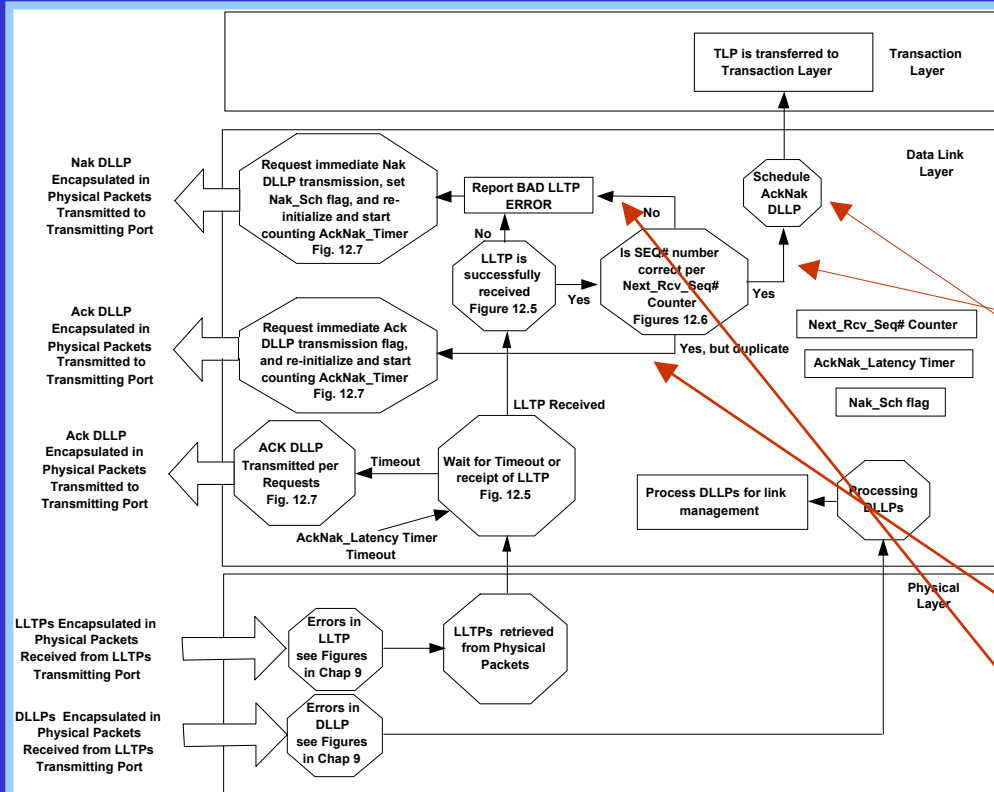
## Flow Control Part 2 ... Protocol for Transmitting and Receiving LLTPs ... continued

- Transmitting LLTPs Flow Control protocol ... continued **Note:** The Figure numbers referenced are the detailed flow charts in the Book.

- Whenever a retry buffer is empty and LLTP is placed into it, the Retry\_Timer is initialized and enabled. If all LLTPs in the retry buffer are acknowledged prior to the timeout of the Retry\_Timer there has been no error in the transmitting of LLTPs via Physical Packets.
- If there are unacknowledged LLTPs in the retry buffer when the Retry\_Timer times out, a Retry\_Timer timeout and all unacknowledged LLTPs in the retry buffer are re-transferred to the Physical Layer for re-transmission.
- If the Retry\_Timer timeout has occurred only a few times, the unacknowledged LLTPs in the retry buffer are simply re-transmitted. If the Retry\_Timer timeout has occurred several times per the Retry\_Num counter, a Retry\_Num# Rollover error has occurred and the following are implemented:

- The link is retrained, the transfer of LLTPs in the retry buffer continues ... OR ...
- If the link can not be retrained the transition is to the L\_Inactive link activity state and no further LLTPs are transferred until Flow Control Initialization is completed.

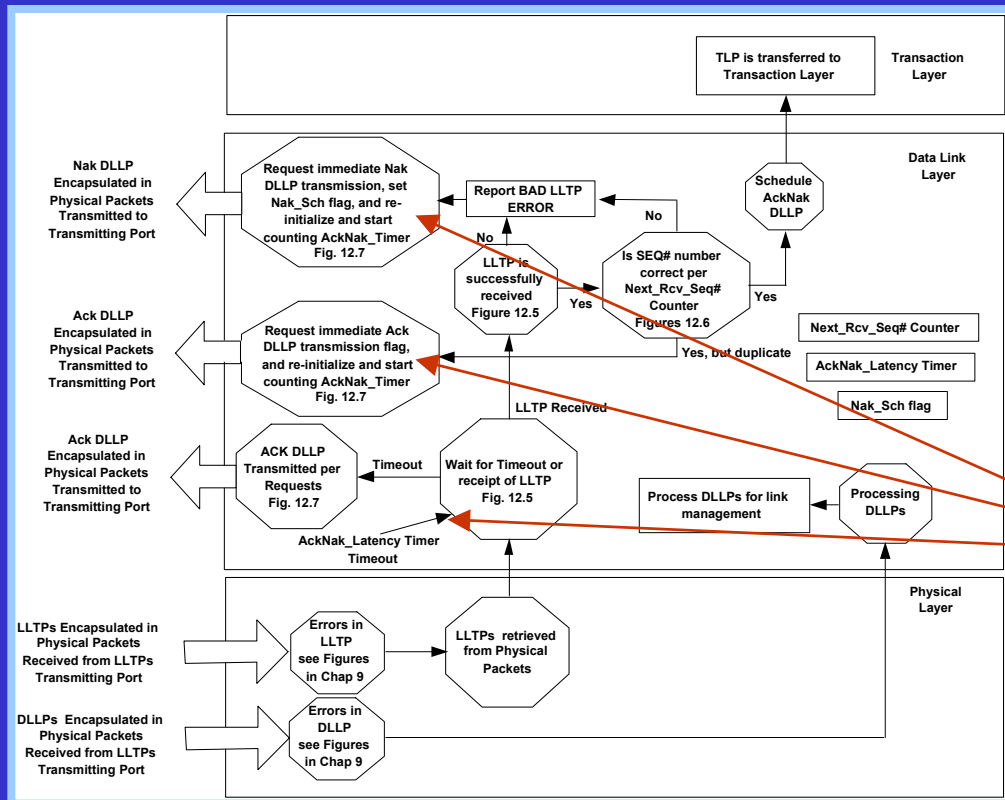
## Flow Control Part 2 ... Protocol for Transmitting and Receiving LLTPs ... continued



- Receiving LLTPs Flow Control protocol **Note:** The Figure numbers referenced are the detailed flow charts in the Book.

- Whenever a Physical Packet is received at the LLTPs' receiving port, LLTPs and DLLPs are extracted. The DLLPs are processed per link management. The LLTP are processed as follows.
- If the LLTP is successfully received in the correct sequence order, the associated TLP is extracted and transferred to the Transaction Layer. Schedule a Ack DLLP to be transmitted.
- If the LLTP is successfully received but is a duplicate SEQ#, the associated TLP is discarded. Schedule an Ack DLLP to be transmitted.
- If the LLTP is not successfully received or received out of order a Nak\_Sch flag is set and schedule a Nak DLLP to be transmitted. BAD LLTP also reported.





## Flow Control Part 2 ... Protocol for Transmitting and Receiving LLTPs ... continued

- Receiving LLTPs Flow Control protocol ... continued **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
- The transmission of a Ack or Nak DLLP results in the AckNak\_Timer being initialized, enabled, and started to transmit another on a periodic basis.

# The Complete PCI Express Reference Topic Group 4 Tutorial

Disclaimer: Intel, Research Tech Inc. and the authors of this tutorial make no warranty for the accuracy or use of the information. No direct or indirect liability is assumed and the right to change any information without notice is retained.

## Design Tools for PCI Express

The PCI Express specification is not organized by clear design topics, does not provide sufficient details to easily master PCI Express, and does not provide in depth illustrations and flowcharts to assist in designing components. Intel recognized that simply summarizing or re-wording the specification as typically done in the other design tools in the industry is insufficient ... more extensive information is are needed.

To provide designers with PCI Express design and implementation information that is easy to master and detailed enough to assist in correct design, two design tools are available:

Six Detailed Tutorials and a new and exhaustively detailed design book.

These design tools focus on Six Quick and Easy Topic Groups which simplify the mastery of PCI Express. They save a designer weeks of trying to unravel the specification and provide the assurance of correct design implementation the first time.

## Design Tools for PCI Express

### The “Book”

The primary design tool that provides total design mastery is *The Complete PCI Express Reference* book written by Edward Solari and Brad Congdon and published by Intel ... referred to as the “Book”.

The Book provides the complete and extensive narrative of detailed figures (over 250), detailed design flow charts, and exhaustive tables for the complete understanding and design assistance in over 1000 pages. The Book can be ordered at [www.amazon.com](http://www.amazon.com) ... ISBN # 0971786194.

### Detailed Tutorials

Six free Detailed Tutorials ... One self paced tutorial for each of the Six Quick and Easy Topic Groups. Each introduces PCI Express information with a narrative that complements detailed figures, flow charts, and tables for each specific Topic Group from the Book. The six free Detailed Tutorials are available at [www.intel.com/intelpress/pciexpresscomplete](http://www.intel.com/intelpress/pciexpresscomplete).

This Detailed Tutorial is of Topic Group 4

Detailed Tutorial: *Power Management & Associated Protocols, Resets, Wake Events, and Link States*

References in the Book: *Chapters 13 to 17*

## PCI Express in Six Topic Groups

### Topic Group 1

**Tutgroup1: *Platform Architecture and Accessing of Resources within Architecture***

**References in the Book: *Chapters 1 to 4***

### Topic Group 2

**Tutgroup2 : *Packets' and Layers' Specifics and Errors***

**References in the Book: *Chapters 5 to 9***

### Topic Group 3

**Tutgroup3 : *Transaction Ordering and Flow Control Part 1 and 2 Protocols***

**References in the Book: *Chapters 10 to 12***

### Topic Group 4

**Tutgroup4 : *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

**References in the Book: *Chapters 13 to 17***

### Topic Group 5

**Tutgroup5 : *Other Hardware Topics***

**References in the Book: *Chapters 18 to 21***

### Topic Group 6

**Tutgroup6 : *Software Considerations***

**References in the Book: *Chapters 22 to 24***

# Power Management & Associated Protocols, Resets, Wake Events, and Link States

## Chapters 13 to 17

### Topic Group 4

The PCI Express specification includes features that permit PCI Express devices and associated links to transition to **lower power states** and to support **Hot Plug** of add-in cards.

**Summary:** The PCI Express platform implements a three of **Reset protocols**, two of which support the transition from a sleep or powered-off condition to powered-on. In addition, there are **wake events** that define how a PCI Express device requests a transition from sleep and to support the **Hot Plug protocol**. These are all part of **Power Management Events** as is the **Slot Power protocol**. Finally, the PCI Express devices and associated links transitioning between different power levels is done via **link states**. The link states also define other possible states for the Physical Layer state machine in the PCI Express devices and associated links.

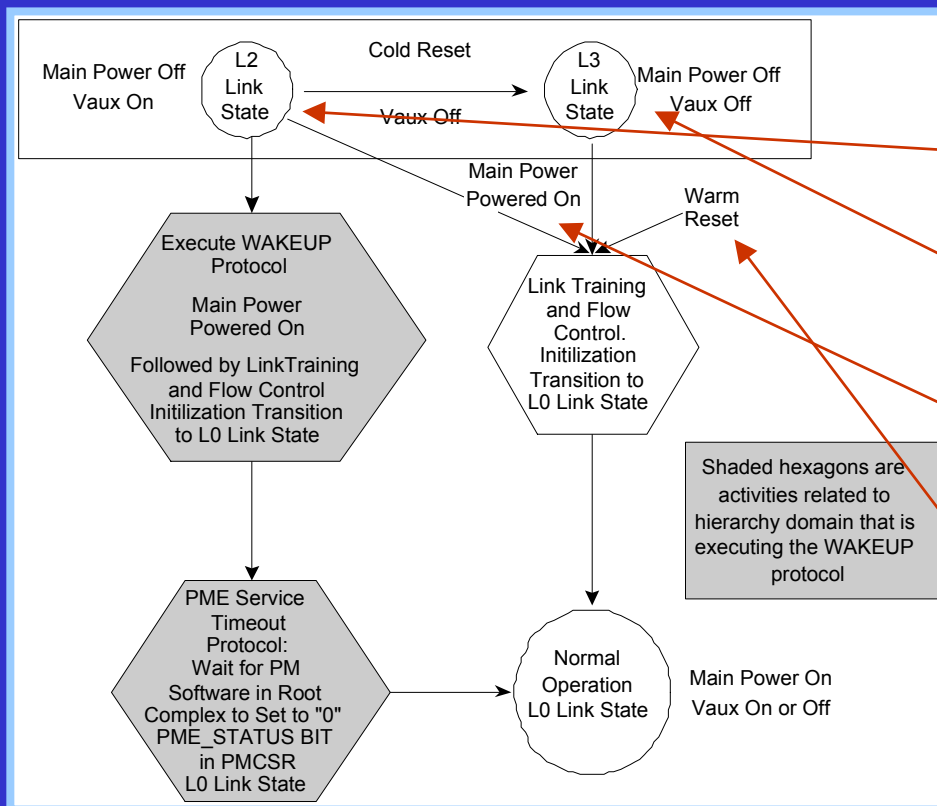
# Chapter 13

## PME Overview, Resets, Wake Events, and Introduction to PM Protocol

## PME Overview

- The Power Management Events (PME) can be viewed in three categories: Power Management protocol, Hot Plug protocol, and Slot Power protocol.
  - **Power Management protocol:** The purpose of the Power Management (PM) protocol is to provide PME software a means to control the power level of the PCI Express devices and the associated links. The element of the PME software related to the PM protocol is Power Management (PM) software. The PM protocol is broken down into the following three tasks:
    - One task: Transition from sleep or powered-off to powered-on for PCI Express devices and associated links. This task includes the resets and wake events
    - Second task: Lower the power consumed by the PCI Express devices and associated links.
    - Third task: Place into sleep or prepare to be powered-off the PCI Express devices and associated links.
  - **Hot Plug protocol:** The purpose of the Hot Plug (HP) protocol is to permit PCI Express add-in cards to be inserted or removed from the platform's slot without powering off the entire platform. The element of the PME software related to the HP protocol is Hot Plug (HP) software.
  - **Slot Power protocol:** The purpose of the Slot Power (SP) protocol is to provide Power Management Event (PME) software a means to control power allocation to add-in cards. The element of the PME software related to the SP protocol is Slot Power (SP) software.
- The subsequent slides will introduce an overview of the PM protocol in order to detail the **wake events** and the different **resets**.
- The final three groups of slides will detail PM , HP and SP protocols, respectively.



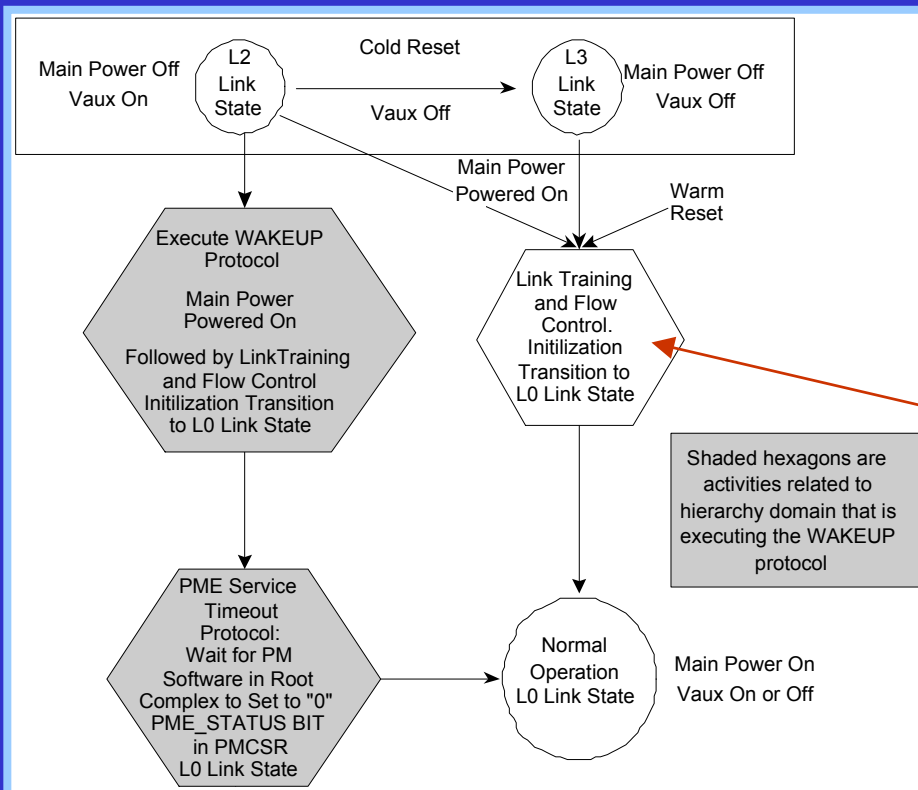


## Resets

- Transition from sleep or powered off to powered on
  - The two PCI Express devices on a specific link are defined in the sleep (L2) when main power is off and Vaux is on. The Vaux is the basic 3.3 volt standby power.
  - The two PCI Express devices on a specific link are defined in the powered off (L3) when main power is off and Vaux is off.
  - L2 and L3 represent specific link states of the Link Training and Status State Machine (LTSSMs) in the PCI Express devices and the associated link.
  - The PCI Express devices and the associated link in the L2 or L3 link states defines a Cold Reset. Cold Reset reflects the value of the PERST# signal line when is asserted when the main power is off.
- Warm Rest is also defined when the PERST# signal line is asserted but the main power is on. It is implementation specific

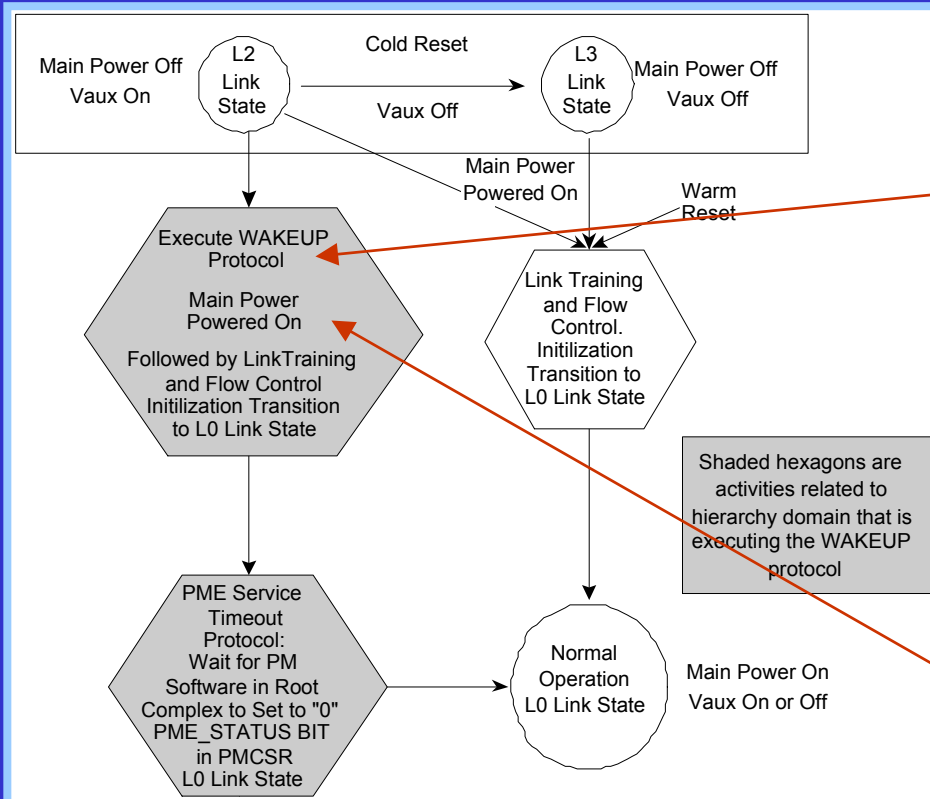
### Resets ... continued

- The Cold and Warm Resets are related to the assertion of the PERST# signal line. In the case of Cold Reset it also includes the application of main power. The Cold and Warm Reset are collectively called the Fundamental Reset.
- The other reset not discussed in the previous slide is Hot Reset and it is not related to main power, other than main power must be on.
- Hot Reset link state places the configured lanes in a specific link into a reset condition. The Hot Reset link state protocol defines two types of ports on a link: Transmitting and Receiving
  - Only a port on a PCI Express device on the upstream side of a link can become the transmitting port. That is, only the Root Complex or switches downstream ports can become a transmitting port. The transmitting port is directly by a higher level to execute Hot Reset.
    - The higher level may be software related. In the case of the switch the higher layer also includes the sensing of Hot Reset on an upstream port with the receipt TS1 OSs with Training Control Bits [2::0] = 001b on *all* configured lanes or upstream port reports Link\_DOWN
  - The other type of port is the receiving port. The receiving port must always be the port on the downstream side of a link. That is, only the upstream ports of the endpoints, switches, and bridges can become receiving ports for a Hot Reset link state. It is directed to the Hot Reset link state by receipt of the two consecutive TS1 OSs with Training Control Bits [2::0] = 001b on *any* configured lanes from the transmitting port.



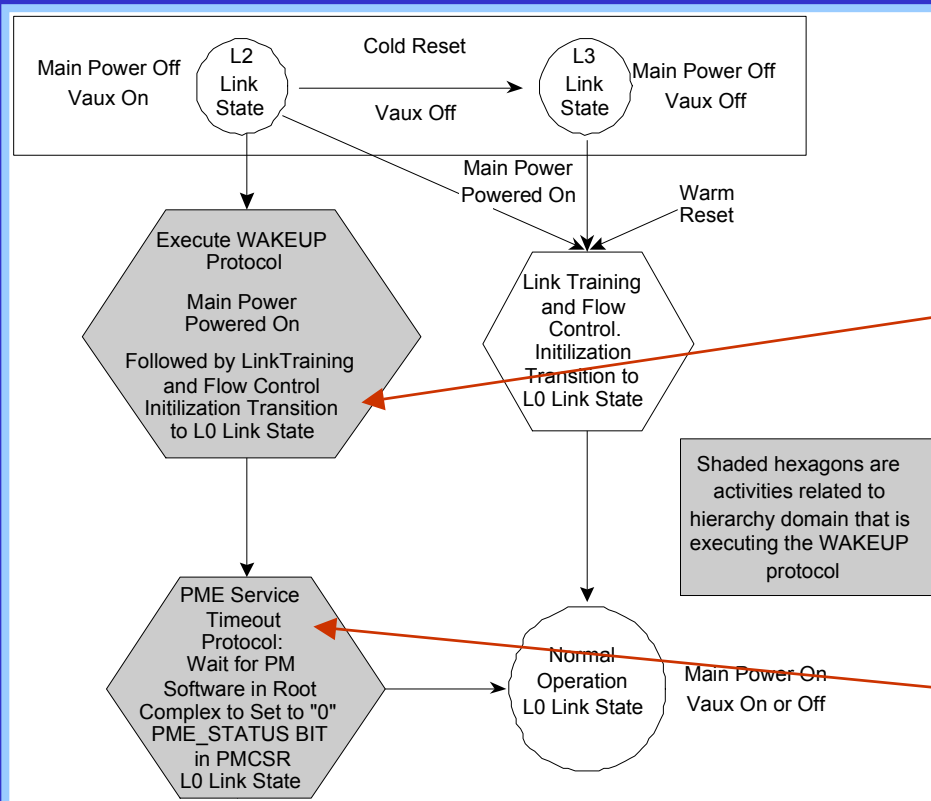
## Wake Events

- Transition from sleep or powered off to powered on
  - To transition from L2 or L3 to L0 only requires main power to be applied (powered on). The application of main power without Wakeup protocol causes Link Training and Flow Control Initialization to be immediately executed by the PCI Express devices.



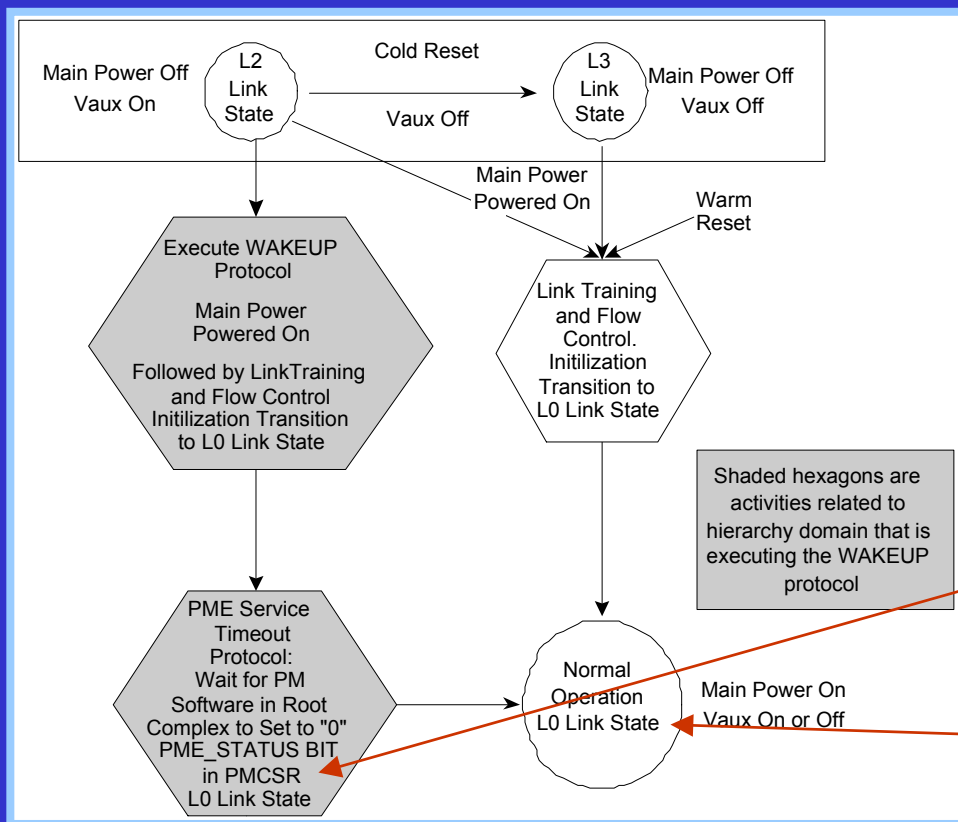
## Wake Events ... continued

- Transition from sleep or powered off to powered on ...continued
  - It is possible for the all or part of a platform to transition from the sleep (L2) to L0 per the request of a PCI Express device downstream to the Root Complex via the Wakeup protocol. The Wakeup protocol consists of several activities
  - First, a PCI Express device asserts the WAKE# signal line or transmits a Beacon. A Beacon is simply a unique electrical pattern on the link signal lines. The eventual destination of the asserted WAKE# signal line or Beacon is the power controller for the PCI Express hierarchy.
  - Second, the power controller applies main power to the PCI Express devices and the associated links. For example, assume that the power controller is at the Root Complex and thus main power is applied entire PCI Express hierarchy attached to a specific the downstream port of the Root Complex.



## Wake Events ... continued

- Transition from sleep or powered off to powered on ...continued
  - Third, once main power is applied the PCI Express devices on each link execute Link Training and Flow Control Initialization protocol.
  - Fourth, the PCI Express device that sourced the WAKE# signal line assertion or the Beacon transmits upstream a message PME\_PM requester transaction packet to the PM software via the Root Complex.
  - In order to insure that the message PM\_PME requester transaction packet reaches the Root Complex and thus the PM software, the PME Service Timeout protocol is applied.



## Wake Events ... continued

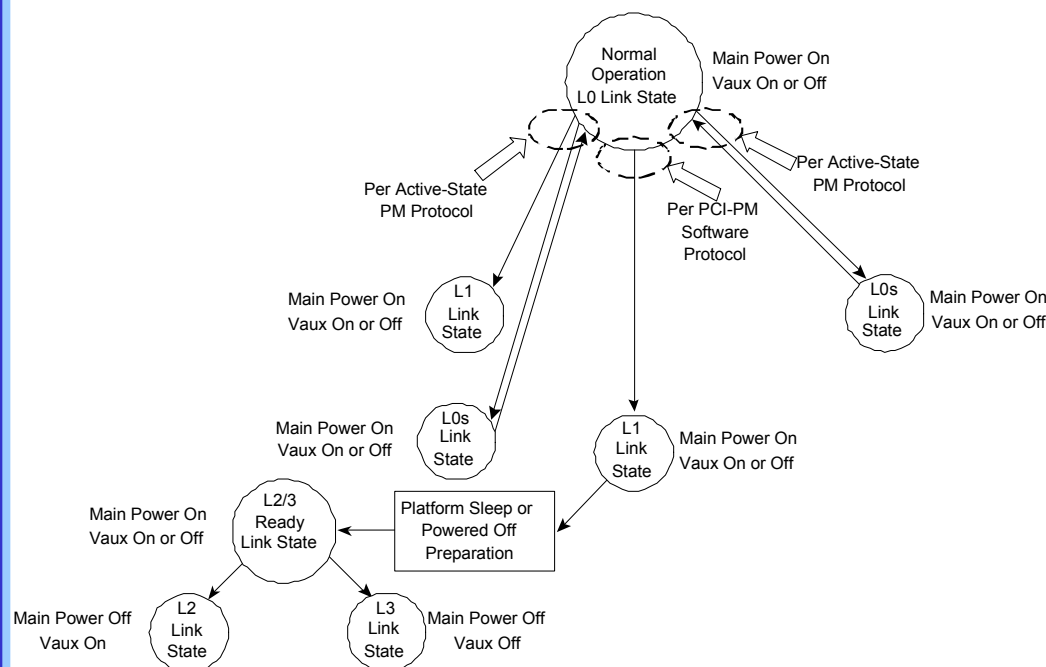
- Transition from sleep or powered off to powered on ...continued
  - The PME Service Timeout protocol insures that the message PM\_PME requester transaction packet is received by PM software in the Root Complex. The transmission of the message PM\_PME requester transaction packet is repeated until the PM software sets to "0" the PME\_STATUS bit in the configuration register block .
  - Upon completion of the PME\_Service Timeout protocol, the PCI Express device is ready for normal operation in the L0 link state.

### Wake Events ... continued

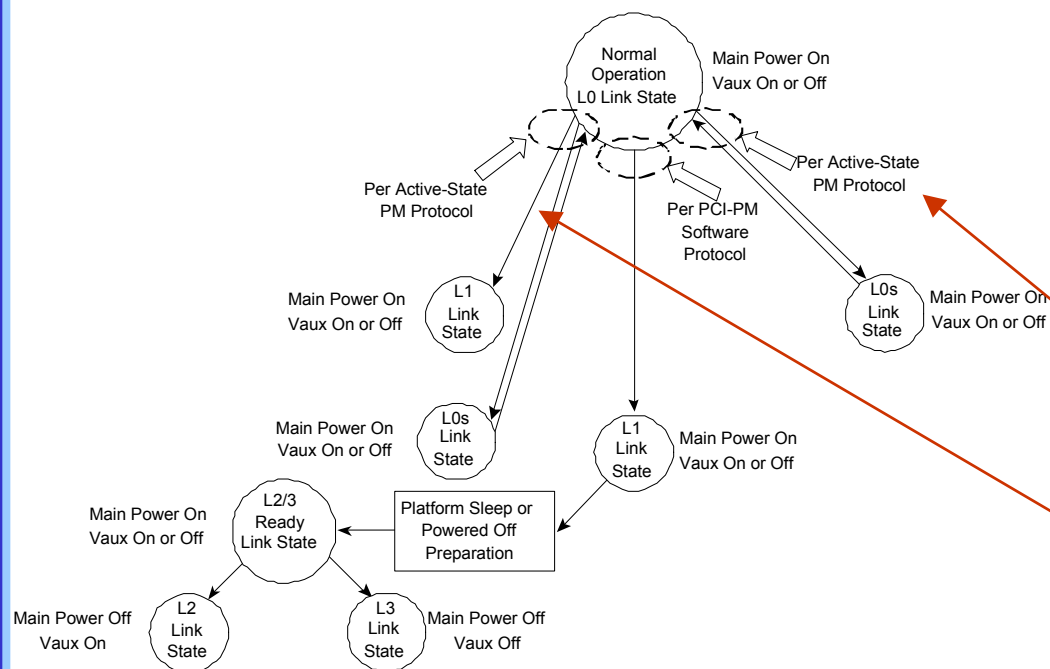
- As discussed in previous slide, the Wakeup protocol is executed by a PCI Express device in sleep (L2) in order to implement a transition to the L0 link state. This is defined as one of the Wake events.
- There is another activity defined as a Wake event called the wakeup event.
  - The wakeup event is defined as part of the Hot Plug protocol. Per the Hot Plug (HP) protocol there are four hardware events: attention button pressed, power fault detected, MSI sensor changed, and add-in card present state changes.
  - In response to these HP hardware events a wakeup event occurs. The wakeup event is simply the sourcing of a message PM\_PME requester transaction packet by the switch. In the case of a Root Complex downstream port it is the “virtual” internal sourcing of this message to itself. This will be discussed in detail in a later slide.
  - During the actual Hot Plug protocol the WAKE# signal line must be “1” (inactive).

## Introduction to PM Protocol

- Lower the power consumed
  - Once the PCI Express device is in the L0 link state it is possible for PCI Express devices and associated links to transition to the lower power link states L0s and L1. L0s provide a low power level with quick transition to the L0 link state. L1 provides a lower power level with a longer transition to the L0 link state
  - In the L0s or L1 link states there is no transfer of Physical Packets across the link and certain entities within the PCI Express devices are operating at reduced power. To transfer Physical Packets across the links requires the L0 link state.

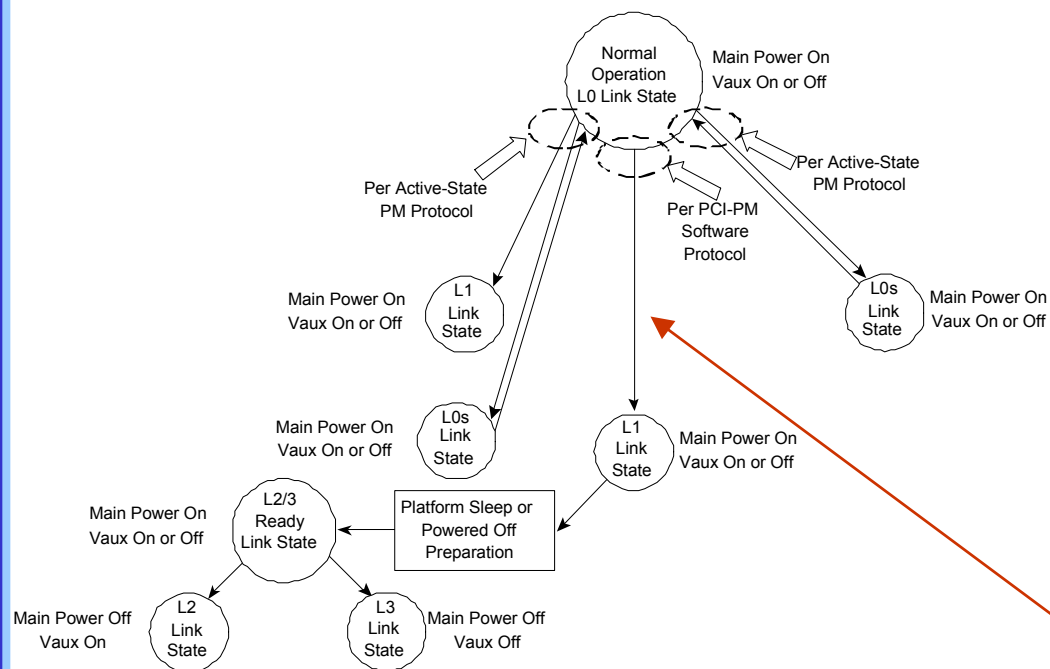






## Introduction to PM Protocol ... continued

- Lower the power consumed ... continued
  - Unique to PCI Express is the Active-State Power Management protocol that supports transitions between the L0 and L0s link states, and between L0 and L1 link states.
    - The transition between the L0 and L0s link states is the simplest and requires no exchange of specific DLLPs. Optionally the transition can occur with exchange of specific DLLPs and TLP.
    - The transition from the L0 and L1 link states requires the exchange of unique DLLPs discussed in later slides.
    - The transition from the L1 to the L0 link state is via the Recovery link state discussed in later slides.

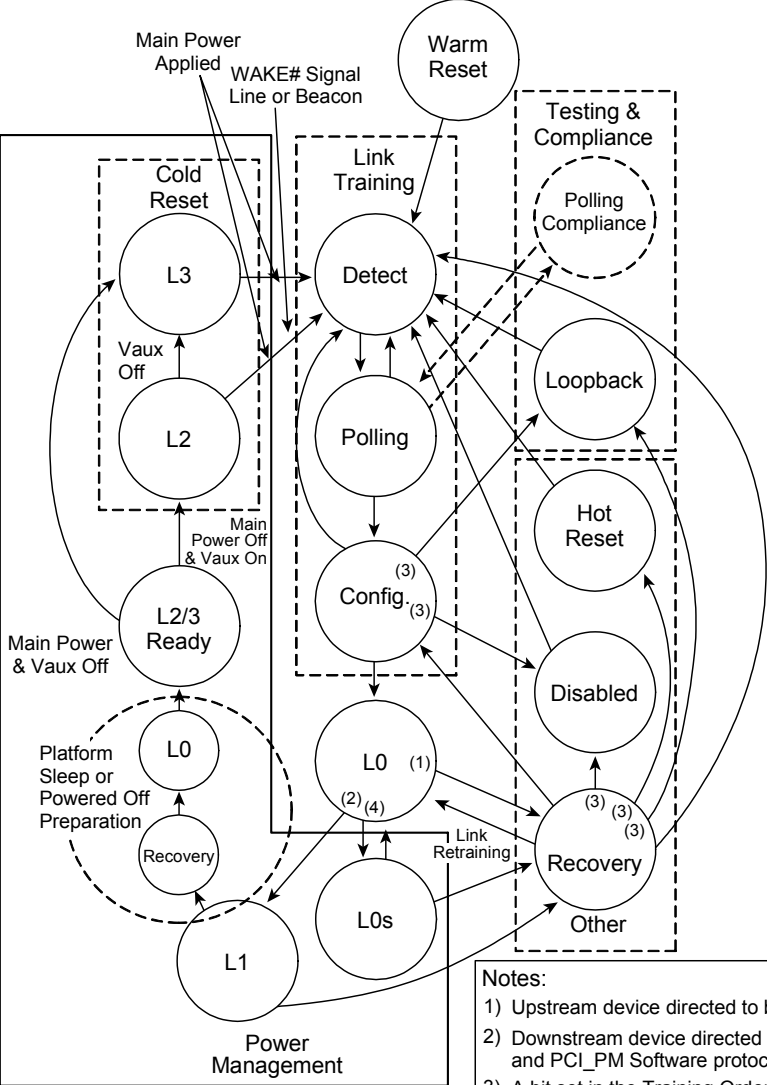


## Introduction to PM Protocol ... continued

- Lower the power consumed ... continued
  - In addition to the PCI Express unique Active-State Power Management protocol, is the sharing of a low power protocol with PCI in the form of the PCI-PM Software protocol.
  - The PCI-PM Software protocol includes the support of PCI power management standards defined in the *PCI Bus Power Management Interface Specification Rev. 1.1* and *Advance Configuration and Power Interface Specification Rev. 2.0*.
  - The PCI-PM Software protocol supports transition from L0 to L1 link state when all functions in the PCI Express devices are programmed in non-D0 link states. The protocol permits the subsequent transition to the L2/3 Ready link state if all functions in the PCI Express devices are programmed into the D3hot state.

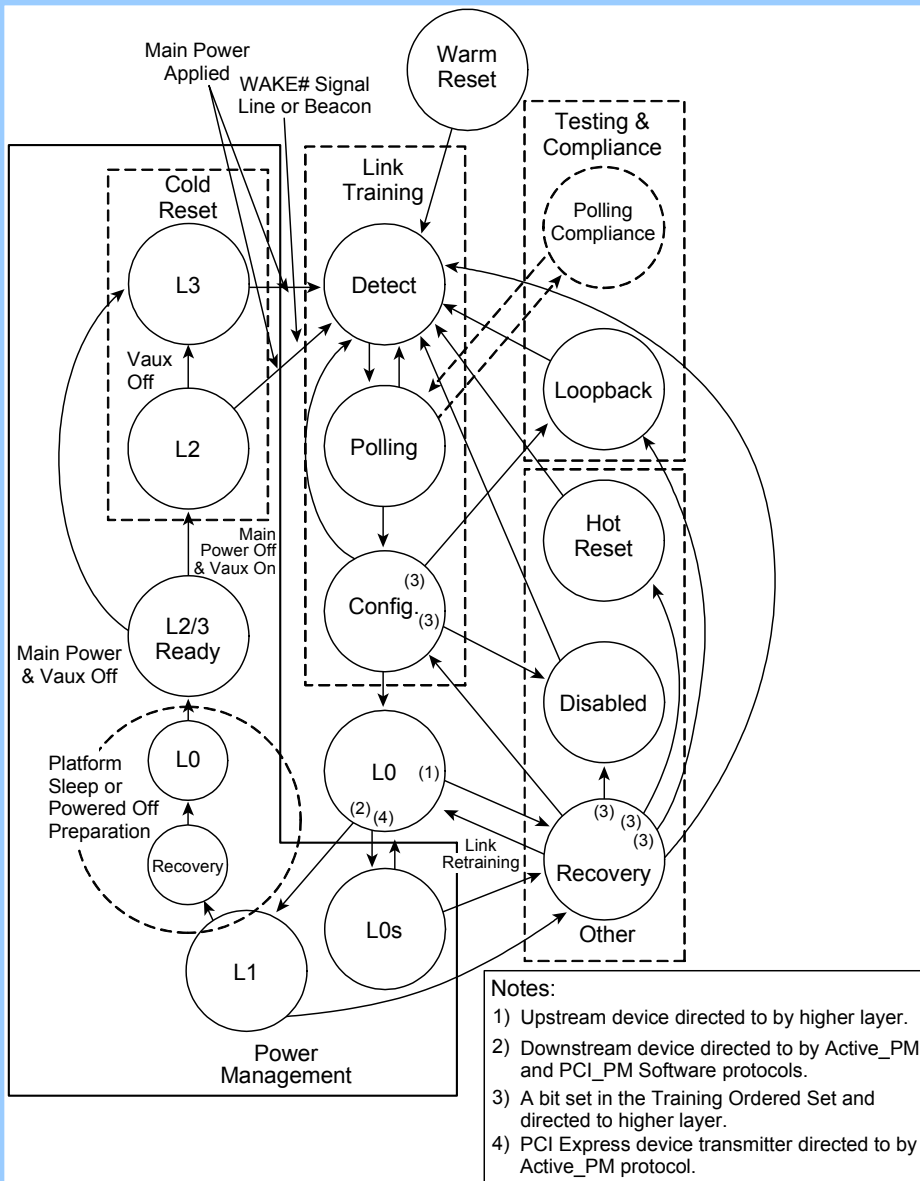
# Chapter 14

## Link States



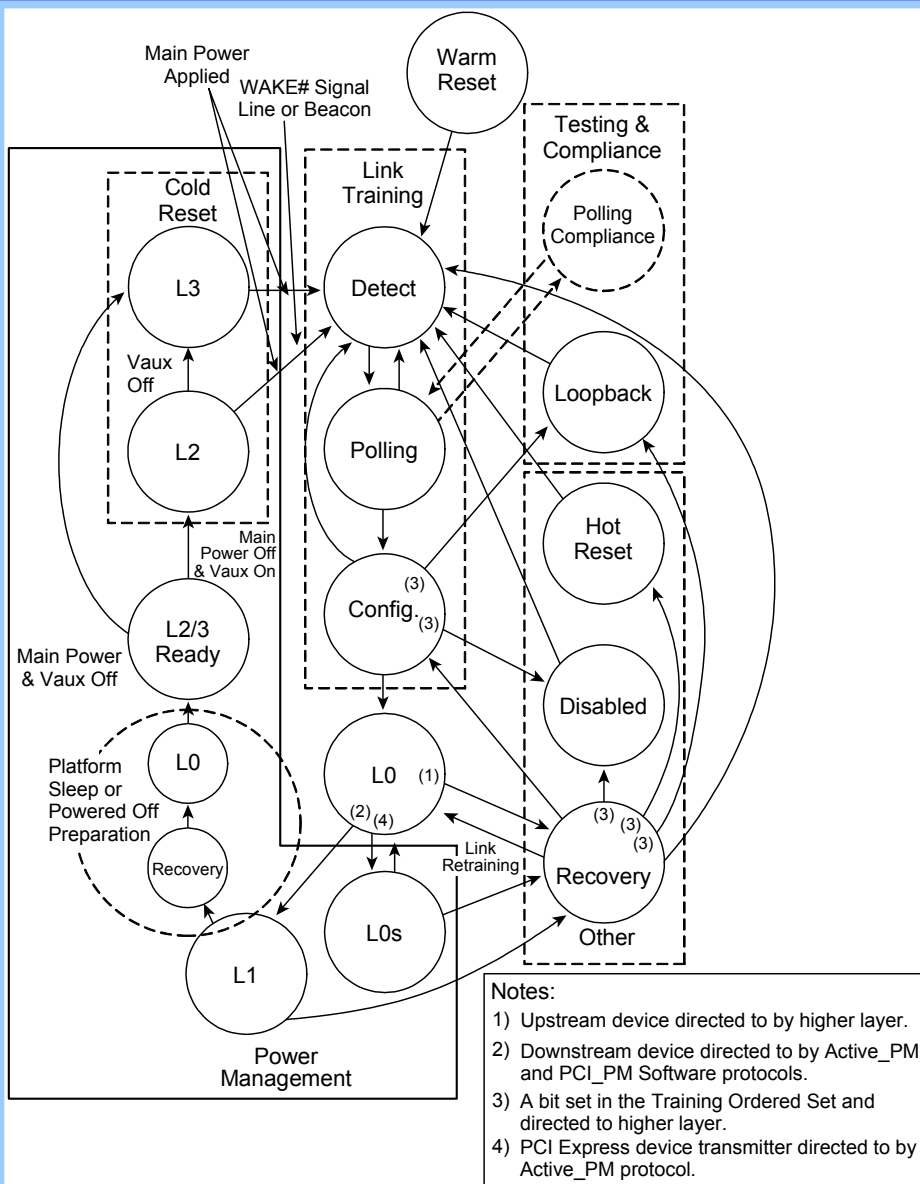
## Link States

- The different link states define the logic states of the Link Training and Status State Machine (LTSSM) in the Physical Layer of each PCI Express device on the link. The links states also define the logic state of the associated link.
- The link states can be grouped into five categories:
  - L0: This is for normal operation when Physical Packets containing both LLTPs and DLLPs are transferred across the link.
  - Power Management: This group consists of link states that define lower power states for the PCI Express devices and the associated link. No Physical Packets are transferred across the link.
  - Link Training: This group consists of link states used to detect the links between two PCI Express devices and configure the appropriate number of lanes on the link.



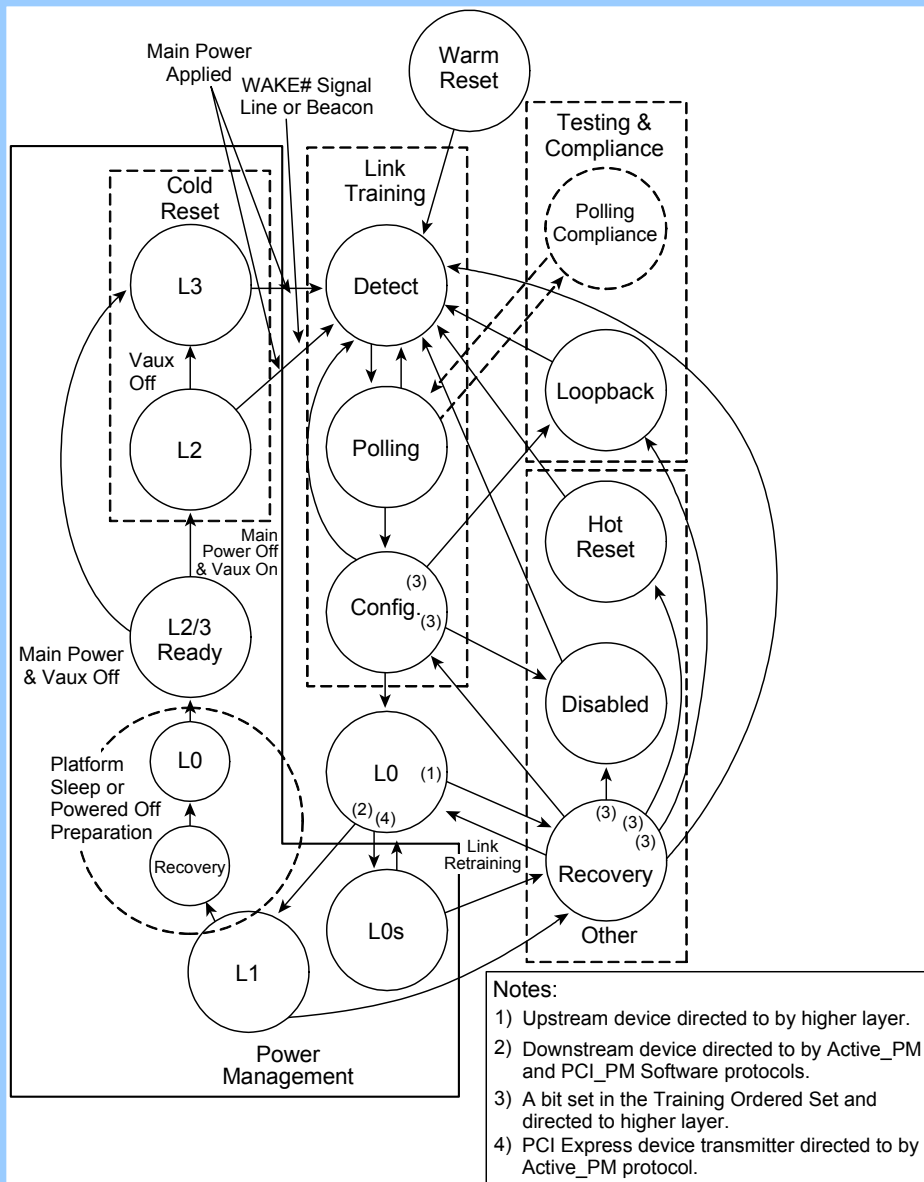
## Link States ... continued

- The link states can be grouped into five categories: ... continued
  - Other: This group is the remaining links states that define a specific operation for the PCI Express devices and the associated link.
  - Testing and Compliance: This a unique group that supports testing (debug) and compliance (validation).
- As previously stated, the link activity states are specific to the Flow Control Initialization protocol. The link states are uniquely different.



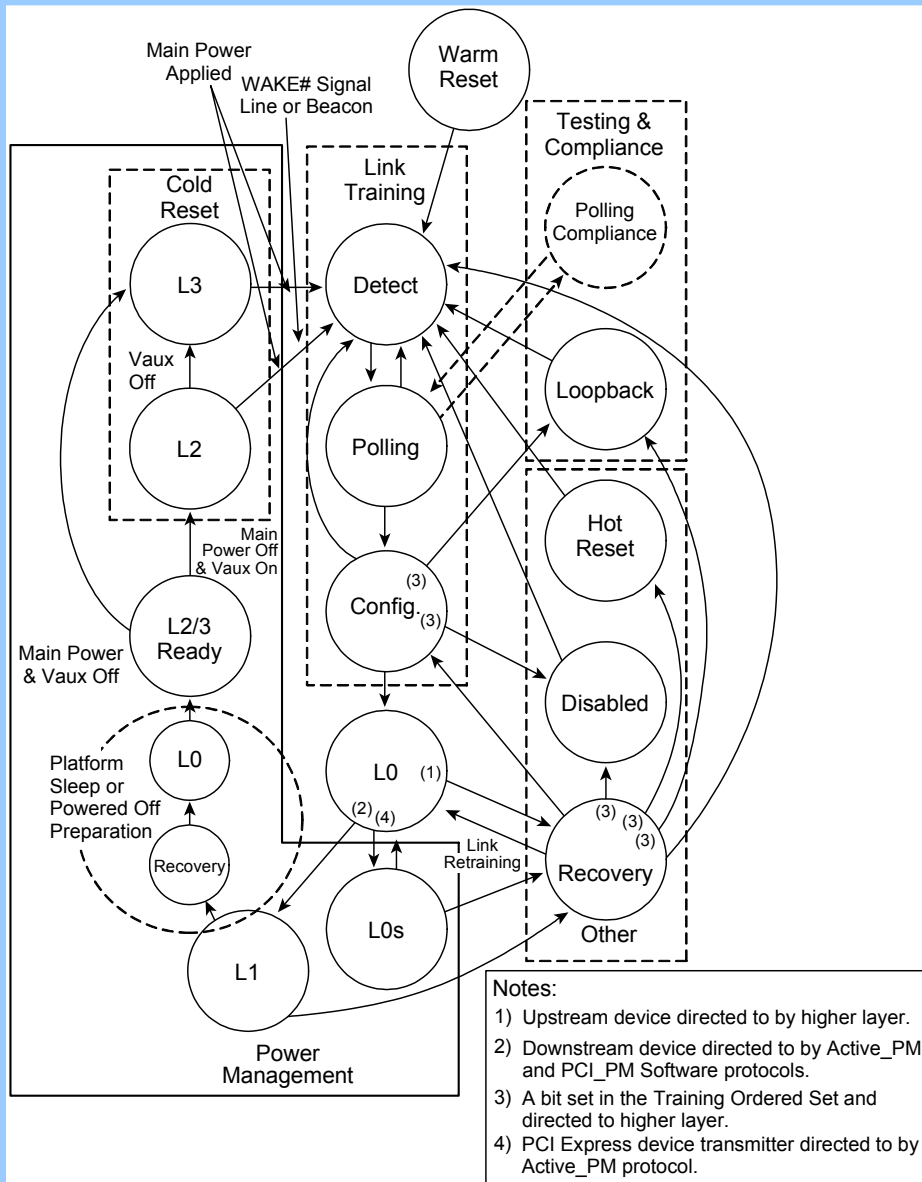
## Link States ... continued

- The transition from link state to link state is very complex and involves link sub-states within each link state. The Book provides detailed Next State Tables to enable designers to understand and design the LTSSM for each PCI Express device. The following slides will provide an overview of the five link states groups and the link states within each group.
- As detailed in the next State tables in the Book, some of the transitions are due to directions by a higher layer. The higher layer directions are activities that occur at the PCI Express device core, Transaction Layer, or Data Link Layer. Some of these will be noted in the following slides, all are detailed in the Book.



## Link States ... continued

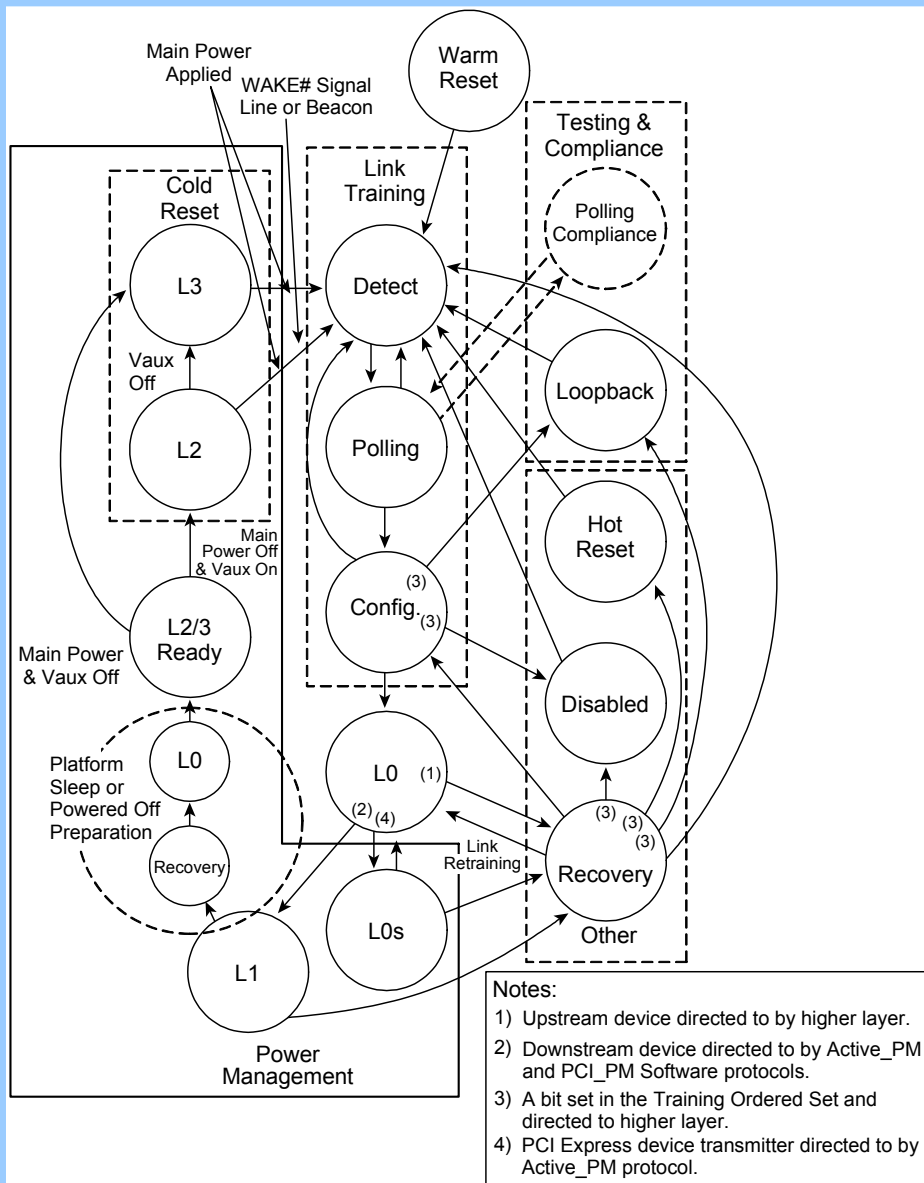
- **L0 Link State:** The PCI Express devices and associated link are in normal operation. Upon entry from the Link Training group the link has been configured and is ready to transmit Physical Packets containing LLTPs (which if course contain TLPs) once the Flow Control Initialization protocol is completed. As discussed in earlier slides the Flow Control Initialization protocol establishes the initial available buffer space at the receiving port of the LLTPs.
  - No Physical Packet containing a LLTP can be transmitted unless the transmitting port knows that sufficient buffer space is available at the receiving port for the LLTPs.
  - The transition from the L0 link state to the L0s and L1 links states of the Power Management group are per the Active-State PM and PCI-PM Software protocol introduced in earlier slides and detailed in later slides.
  - The transition from the L0 link state to the Recovery link state of the Other group occurs via a higher layer direction by software setting the Retrain bit in the configuration register block.



## Link States ... continued

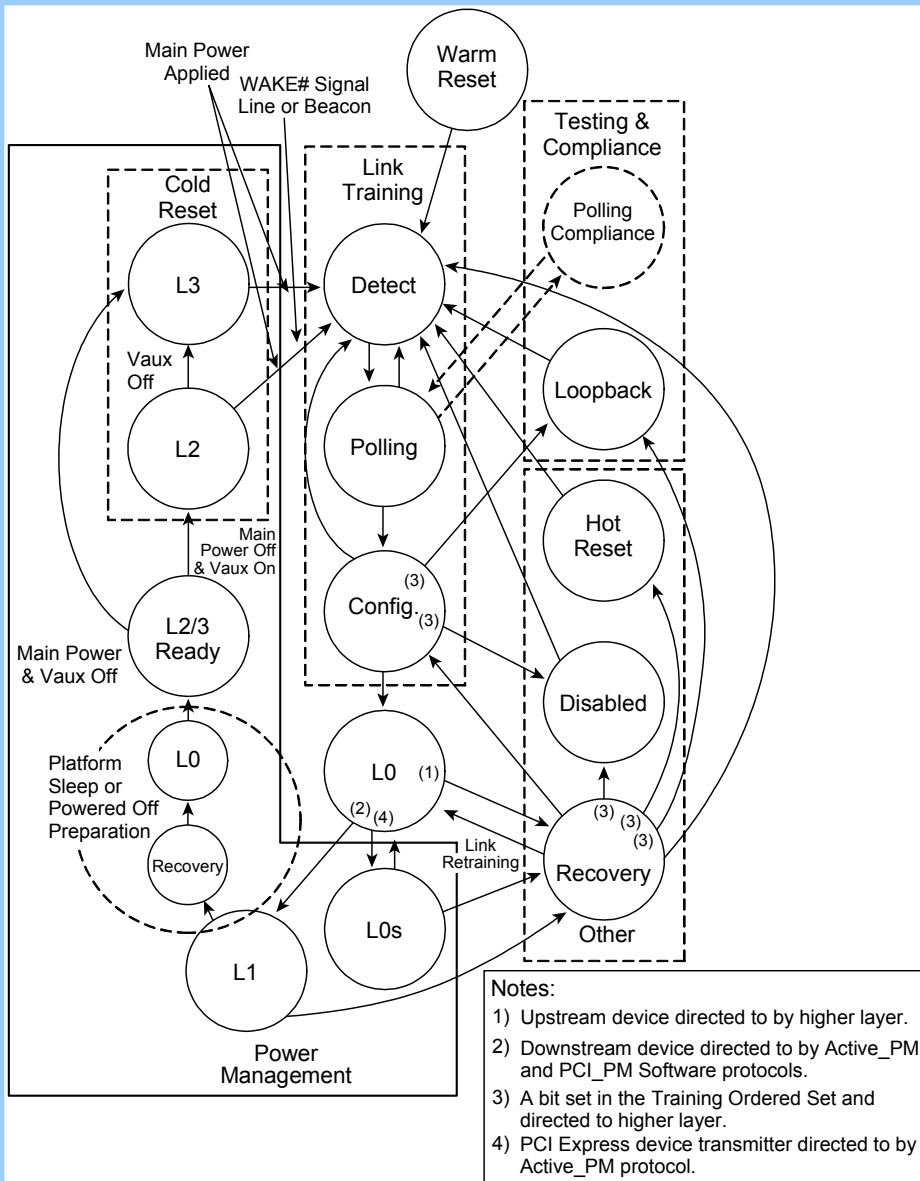
- **Power Management Group:** Once the PCI Express devices and associated link enter this group either lower power will be consumed for a brief period or it is in preparation for turning off main power.
  - As discussed in earlier slides and will be detailed in later slides, the Active-State PM protocol will cause the transition to the L0s or L1 link state when there are no Physical Packets to transfer.
  - As discussed in earlier slides and will be detailed in later slides, the PCI-PM Software protocol will cause the transition to the L1 link state when functions within the PCI Express devices are programmed into specific “D” states.





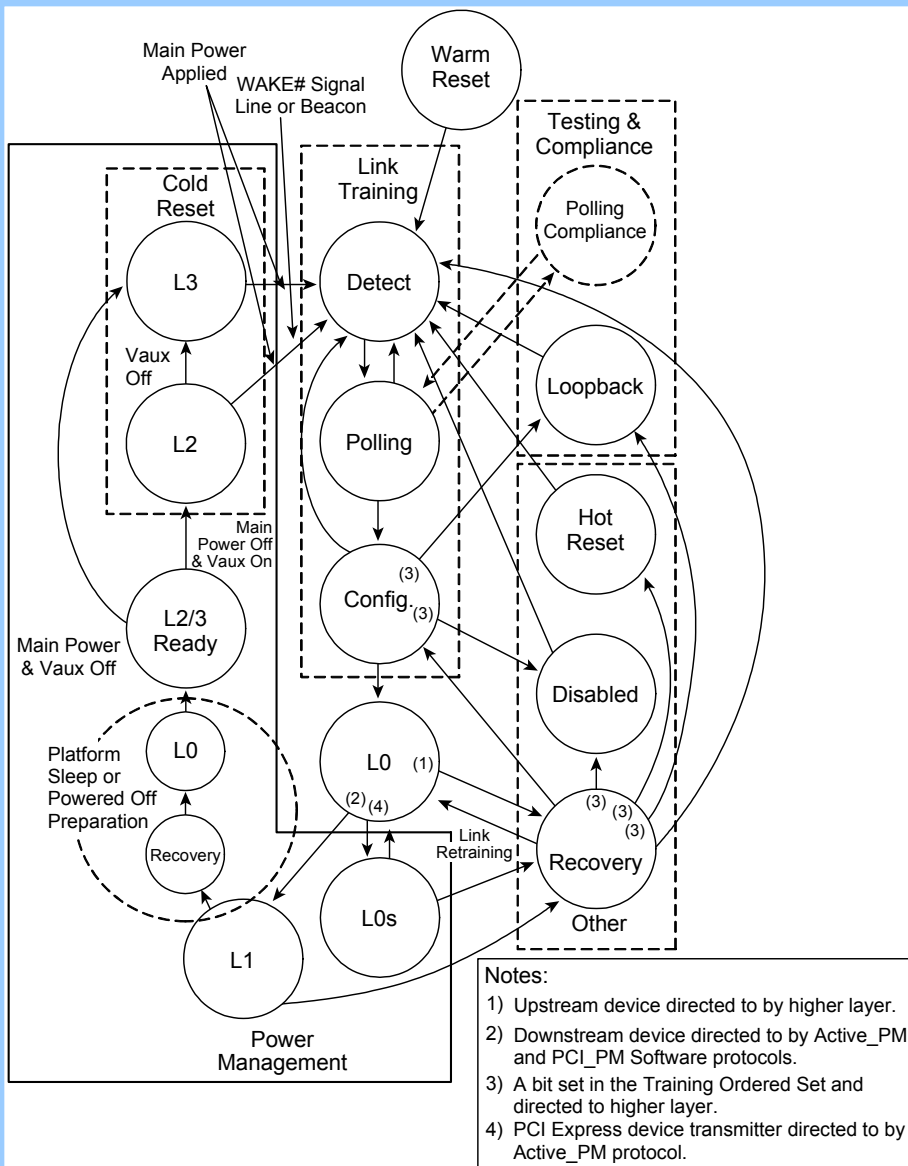
## Link States ... continued

- Power Management Group: Once the PCI Express Devices and associated link enter this group either lower power will be consumed for a brief period or it is in preparation for turning off main power ... continued
  - As discussed in earlier slides, the transition from the L1 link state to the L2/3 Ready link state requires a temporary transition through the L0 link state via the Recovery link states. This is defined as the “Platform Sleep or Powered Off Preparation” and is detailed in the book.
  - Once in the L2/3 Ready link state the transition to sleep (L2) occurs if main power is turned off and Vaux remains on.
  - As detailed in earlier slides, the transition from the L2 or L3 link states of the Power management group to the Link Training group is due to the Wakeup protocol (WAKE# signal line or Beacon) or simply by the application of main power.



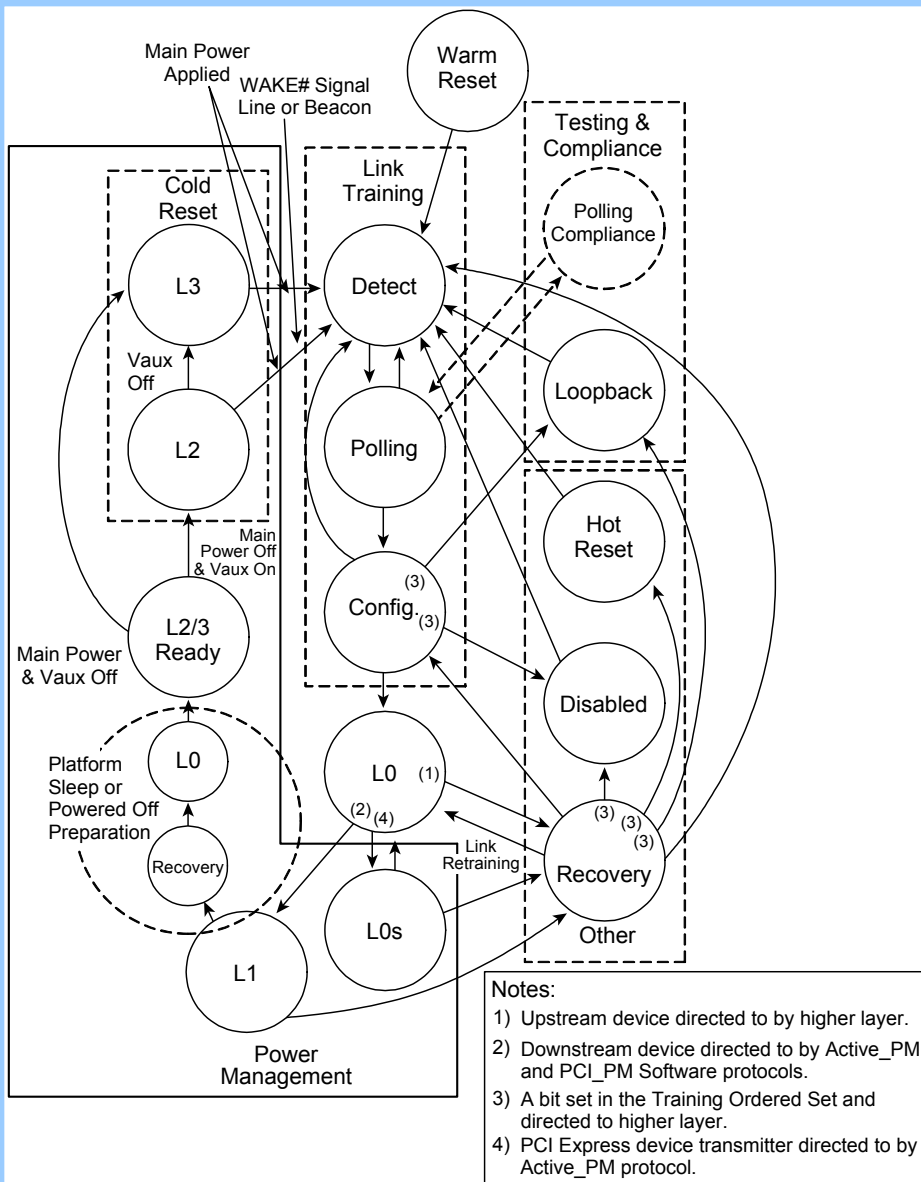
## Link States ... continued

- **Link Training Group:** Once the PCI Express devices exit from a Cold Reset or Warm Reset, Recovery, Loopback, Hot Reset, or Disabled link state the structure of the link between these PCI Express devices is unknown. Link Training determines the structure of the link between two PCI Express devices via the transition through three link states as follows:
  - **Detect Link State:** This is the entry point of the Link Training Group. The purpose of this link state is for a port's transmitters to detect the presence of receivers on the other end of the link on lanes common to both PCI Express devices on the link.



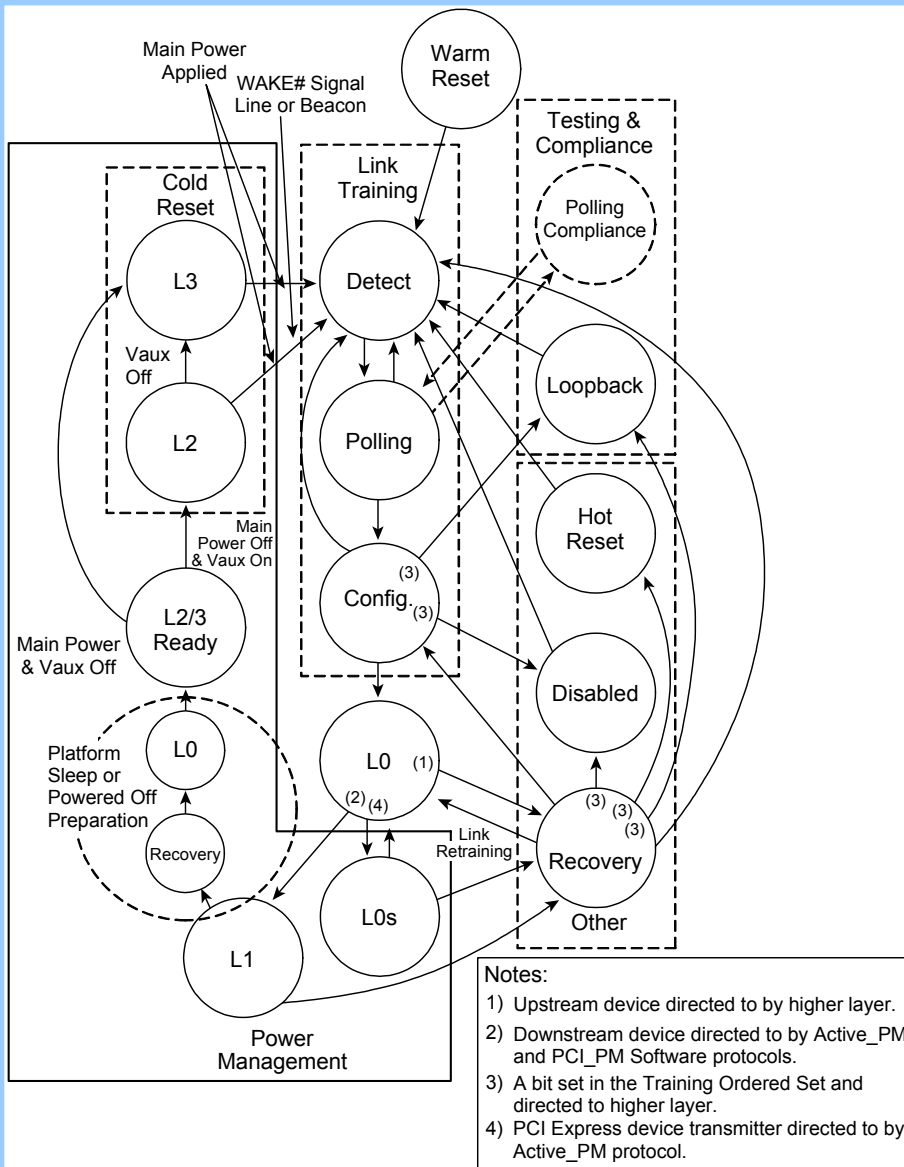
## Link States ... continued

- Link Training Group ... continued
  - Polling Link State: The purpose is to establish bit and symbol lock on the all detected un-configured lanes defined by the Detect link state. It also establishes lane polarity inversion and data bit rate for these detected un-configured lanes. One of the polling link state sub-states is also used for compliance testing with test equipment.
    - The integration of the reference clock into the bit stream of the Physical Packets requires the extraction of this clock at the other end of the link. The extraction is done by a phase lock loop (PLL). The PLL must establish a lock on the bits and the associated symbols of the Physical Packets and other symbols transmitted across the link.
    - The lane polarity inversion permits the differentially driven signal line pair to be routed in any fashion through the layers of printer circuit boards without consideration of polarity. The receivers will adjust to the polarity.
    - The transfer rate of the individual bits of the symbols that comprise the Physical Packets. The initial value is 2.5 Gb per second. In later revisions of the PCI Express it is proposed that higher bit rates will be possible. During the Polling link state these higher bit rates will be established.



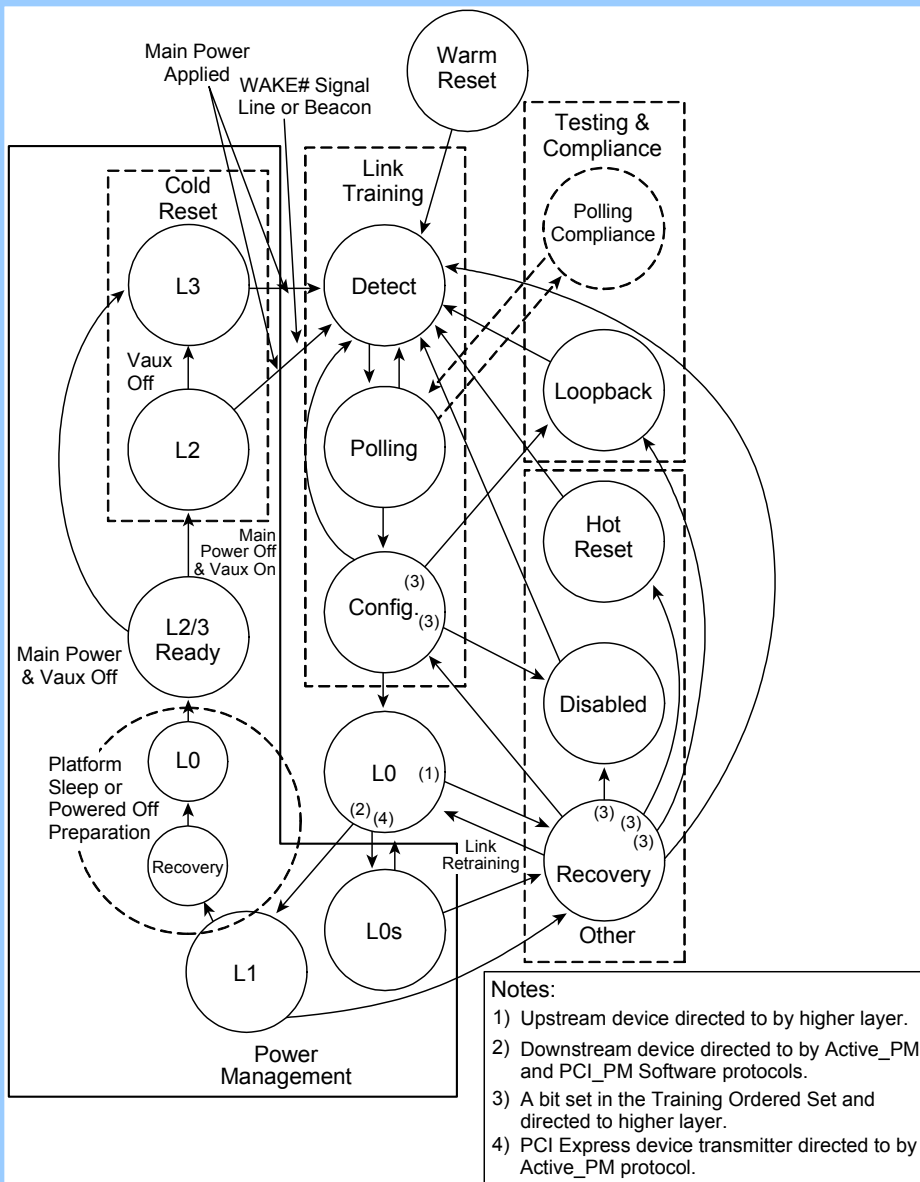
## Link States ... continued

- Link Training Group ... continued
  - Configuration Link State: Prior to entering the Configuration link state, the link has achieved bit and symbol lock, lane reversal polarity has been completed, and the highest common data bit rate has been established on *ALL* detected un-configured lanes. The next step is to configured set of lanes into a configured link.
    - Once link configuration is completed a LINK number (LINK#) has been established and a LANE number (LANE#) for each configured lane of the link has been established.
    - To establish the LINK# and LANE# the LTSSMs of the two PCI Express devices have exchanged Training Set 1 and 2 (TS1 and TS2) Ordered Sets. The TS OSs from the upstream device (USD) on the link provides the LINK# on all the lanes it can potentially configure. The downstream device (DSD) replies with the returning the LINK# on all lanes it can configure.
    - The USD and DSD continue the exchange of TS OSs with the LINK# and a range of LANE#s to determine the lanes that can be configured.



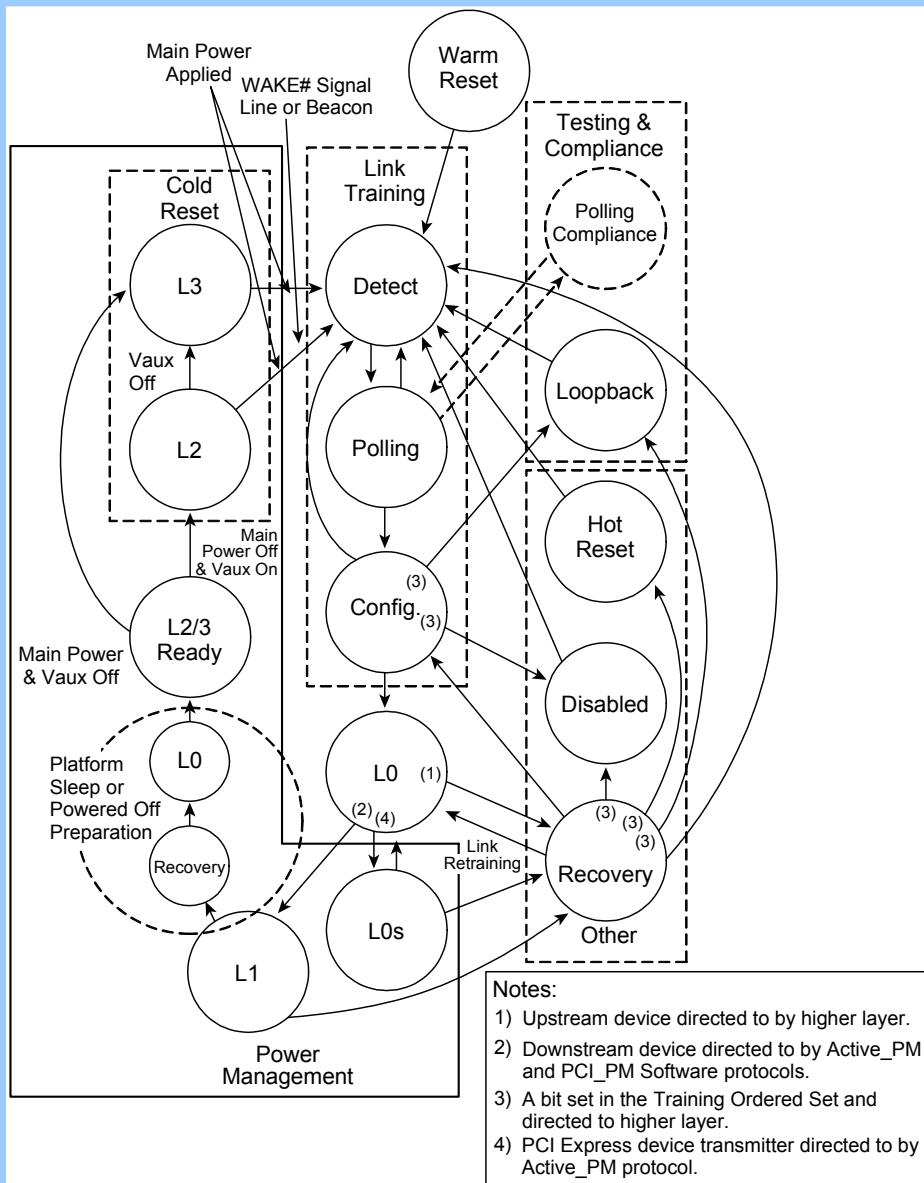
## Link States ... continued

- Link Training Group ... continued
  - Configuration Link State: .. Continued
    - Transition to the L0 link state occurs with successful establishment of a configured link of configure lanes.
    - Transition to the Loopback link state is used for validation and debug with test equipment, and will be discussed in a later slide.
    - Transition to the Disable link state occurs when the Disable bit is set by software.



## Link States ... continued

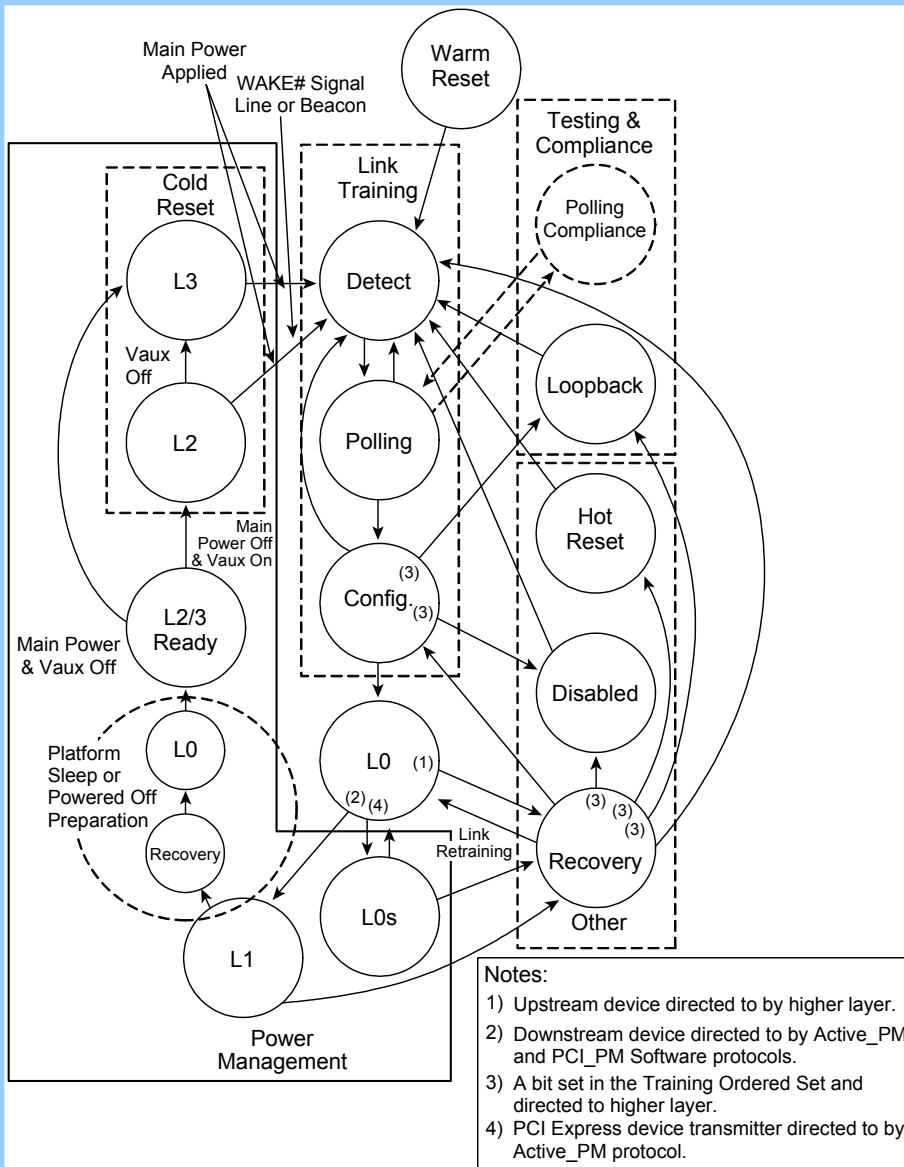
- Other Group: This group contains three distinct link states.
  - Recovery: The Recovery link state has four uses: First, as a means to transition to the Hot Reset, Loopback, Disable, Detect, and Configuration link states from the L0, L1, and L0s link states. Second, as a means to transition to the L0 link state from the L1 link state and under certain conditions from the L0s link state; Third, as a means to reestablish bit and symbol lock (Link Retraining); and Fourth, to complete lane-to-lane de-skew.
    - As noted in the figure the transition to Detect, Disabled, or Loopback link states is possible from the Configuration link state. Once the transition from the Configuration link state to the L0 link state, the only possible path to transition to these states is via Recovery link state. In the case of Loopback and Disabled, higher layer in L0 link state transitions the link to the Recovery link state and subsequently to the Disabled and Hot Reset link states. It is also possible for the higher layer to direct the link from the L0 to the Recovery link state for Link Retraining and subsequently to Detect link state if Link Retraining is not successful.



## Link States ... continued

- Other Group: ... continued
  - Recovery: ... continued
    - The Part of the Recovery link state is the execution of the Link Retraining which reestablishes the bit and symbol, reestablishes the lane-to-lane skew, and sets a new N\_FTS value for use of future L0. If Link Retraining is not successful it is possible to transition to the beginning of the Link Training Group (Detect link states) and “begin from scratch”. As previously stated the Recovery link state can be entered from the L0 link state per higher layer request. The lower power requirements of the L1 and L0s link states independently must transition to the Recovery link state to execute Link Retraining.

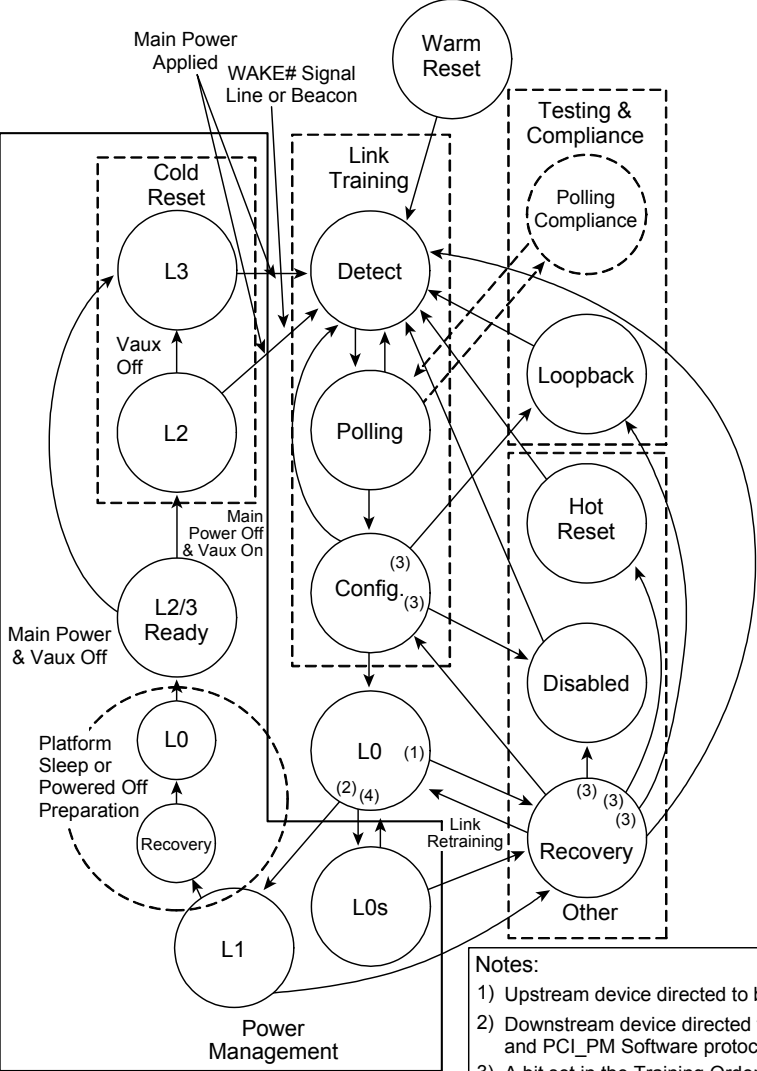




## Link States ... continued

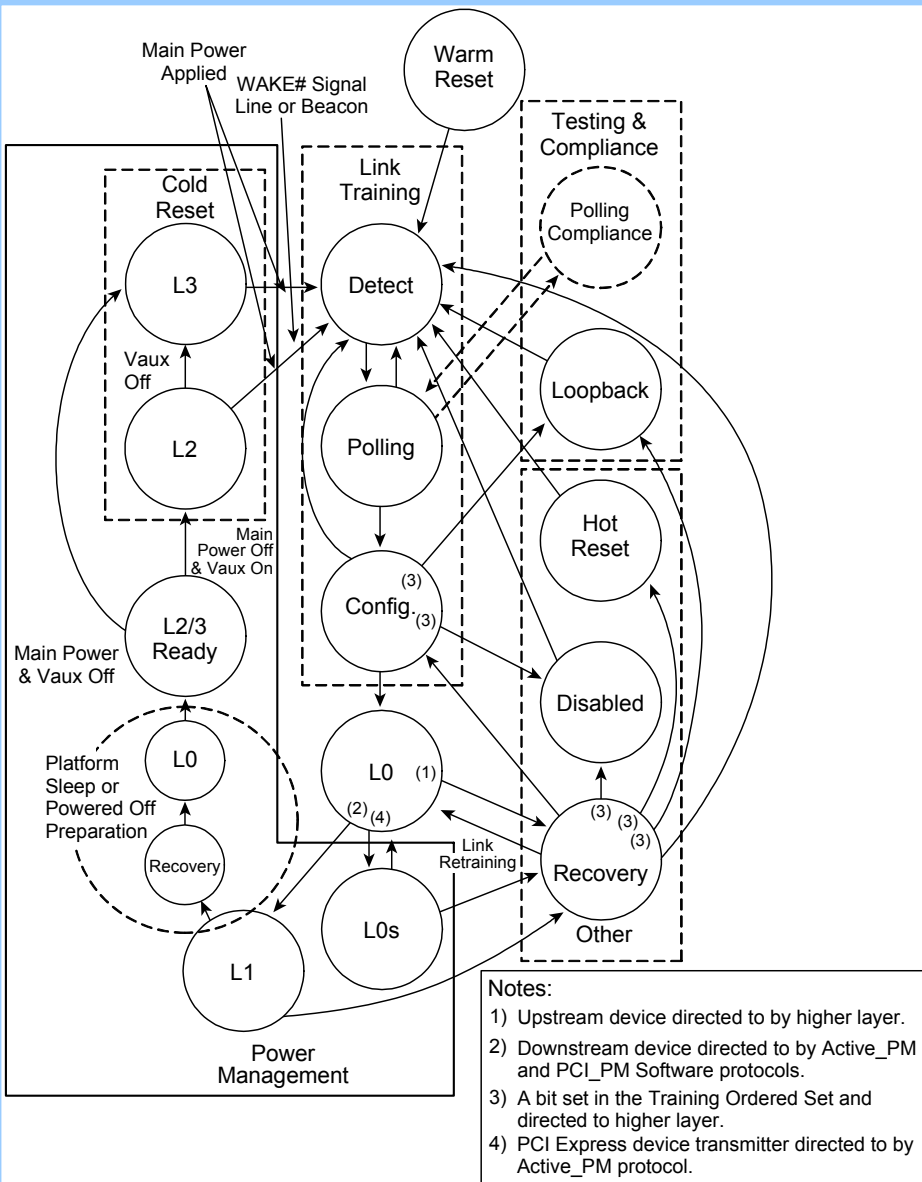
- Other Group: continued
  - Disabled: The transition to the Disabled link state is either from the Recovery link state or the Configuration link state. The Disabled link state can only be requested by downstream ports of the Root Complex or switches (Upstream Device ... USD). The upstream ports of switches, endpoints, and bridges must monitor for the transition to the Disable link state. One application for the Disabled link state is to permit the PCI Express devices associated with a slot to implement the Hot Plug protocol. The USD requests the Disabled link state by transmitting TS1 OSs with the Training Control Bits [2:0] = 010b as directed by higher layer. The transition of the LTSSMs and the associated transmitters and receivers transition on each end of the link is summarized in Tables 14.38 and 14.39 in the Book.





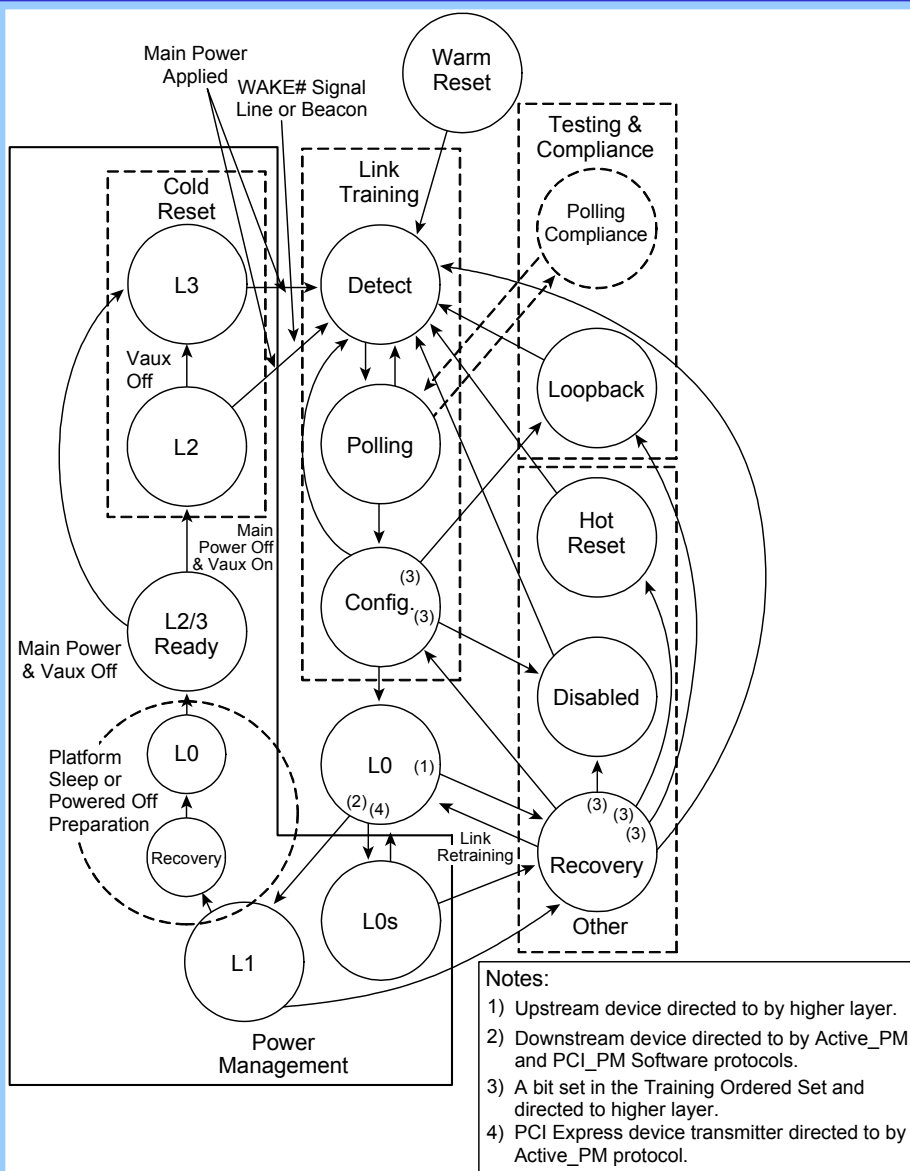
## Link States ... continued

- Other Group: continued
  - Hot Reset: The Hot Reset link state places the link into a reset condition. The transition to the Hot Reset link state is only from the Recovery link state by a higher layer. The Hot Reset link state can only be requested by downstream ports of the Root Complex or switches (Upstream Device ... USD). The USD requests the Reset link state by transmitting TS1 OSs with the Training Control Bits [2:0] = 001b as directed by higher layer. The upstream ports of switches, endpoints, and bridges must monitor for the transition to the Hot Reset link state. The transition of the LTSSMs and the associated transmitters and receivers transition on each end of the link is summarized in Tables 14.40 and 14.41 in the Book.



### Link States ... continued

- Testing and Compliance Group: This group contains two distinct link states.
  - Loopback Link State: The purpose of this link state is to provide a mechanism for a PCI Express device to be tested and any faults to be detected. The transition to the Loopback link state is either from the Recovery link state or the Configuration link state. Per the Loopback protocol, a Loopback Master and a Loopback Slave must be established. The Loopback Master can be established on the downstream ports of the Root Complex or switches. Thus the upstream ports of switches, endpoints, and bridges must monitor for the transition to the Loopback link state and be established as the Loopback Slave. The Loopback Master can also be established on upstream ports of switches, endpoints, and bridges. Thus the downstream ports of the Root Complex or switches monitor for the transition to the Loopback link state and be established as the Loopback Slave.



## Link States ... continued

- Testing and Compliance Group ... continued
  - Polling.Compliance: This link sub-state is used to test the transmitters of a PCI Express device for compliance with the voltage and timing requirements of the PCI Express specification.

- ## Link States ... continued
- Testing and Compliance Group ... continued
    - Polling.Compliance: This link sub-state is used to test the transmitters of a PCI Express device for compliance with the voltage and timing requirements of the PCI Express specification.

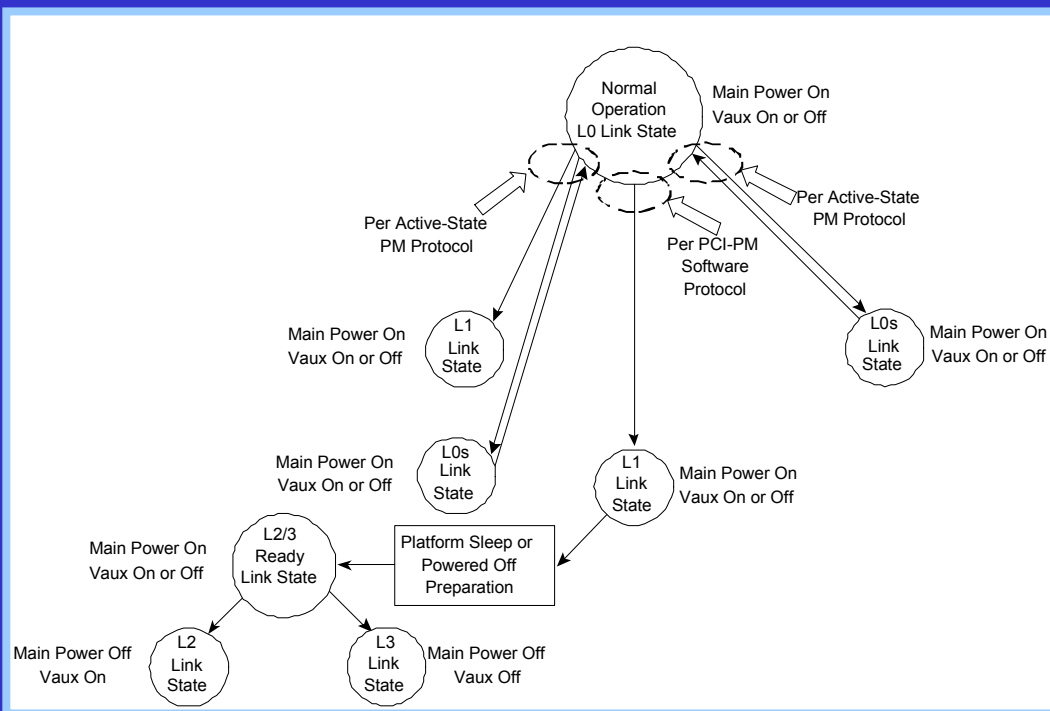
Notes:

- Notes:**
- 1) Upstream device directed to by higher layer.
  - 2) Downstream device directed to by Active\_PM and PCI\_PM Software protocols.
  - 3) A bit set in the Training Ordered Set and directed to higher layer.
  - 4) PCI Express device transmitter directed to by Active\_PM protocol.

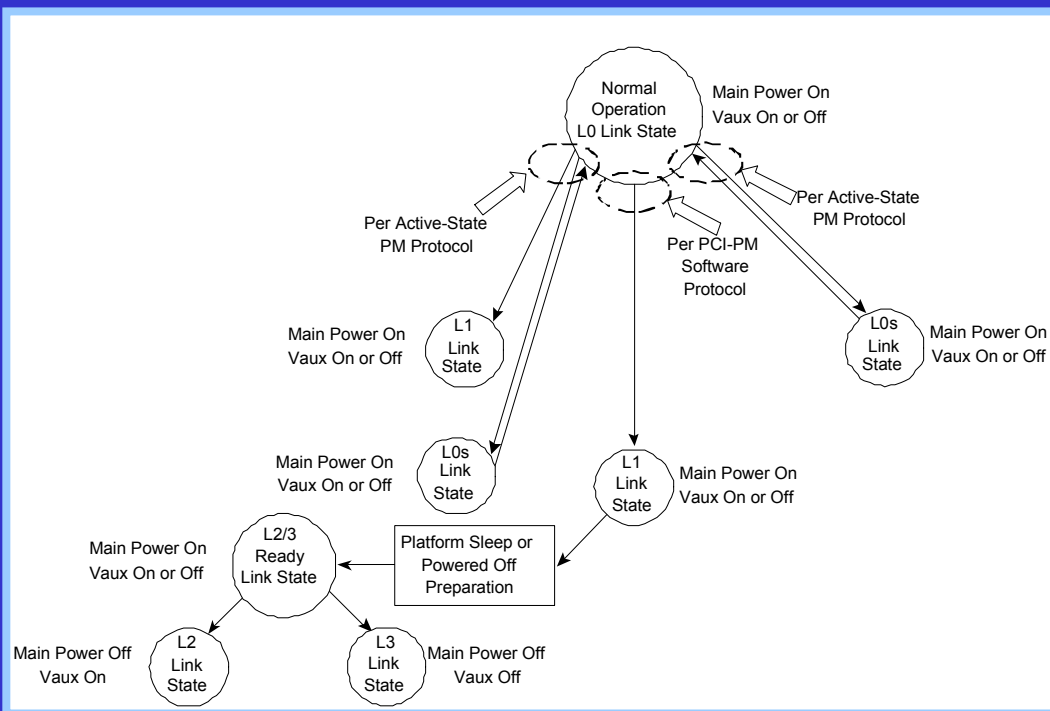
# Chapter 15

## Power Management

## Power Management

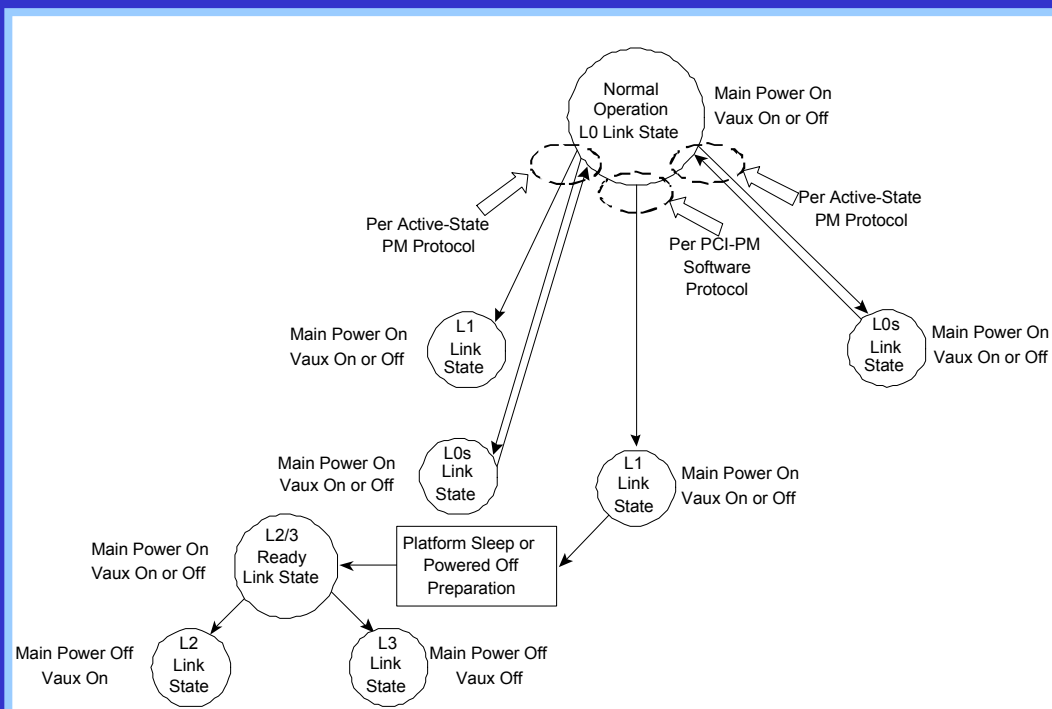


- As discussed in earlier slides, the LSSTMs within the PCI Express devices and the associated link operate normally in the L0 link state. In the L0 link state Physical Packets containing LLTPs and DLLPs transverse the link.
- As previously discussed under certain conditions the PCI Express devices and associated link transition to lower power link state (L0s or L1). Per other conditions the PCI Express devices and the associated links will transition of a link state (L2/3 Ready) in preparation to remove of all power or just the main power.
- As previously stated, the transition to lower power states are implemented by the Active-State PM protocol or the PCI-PM Software protocol.



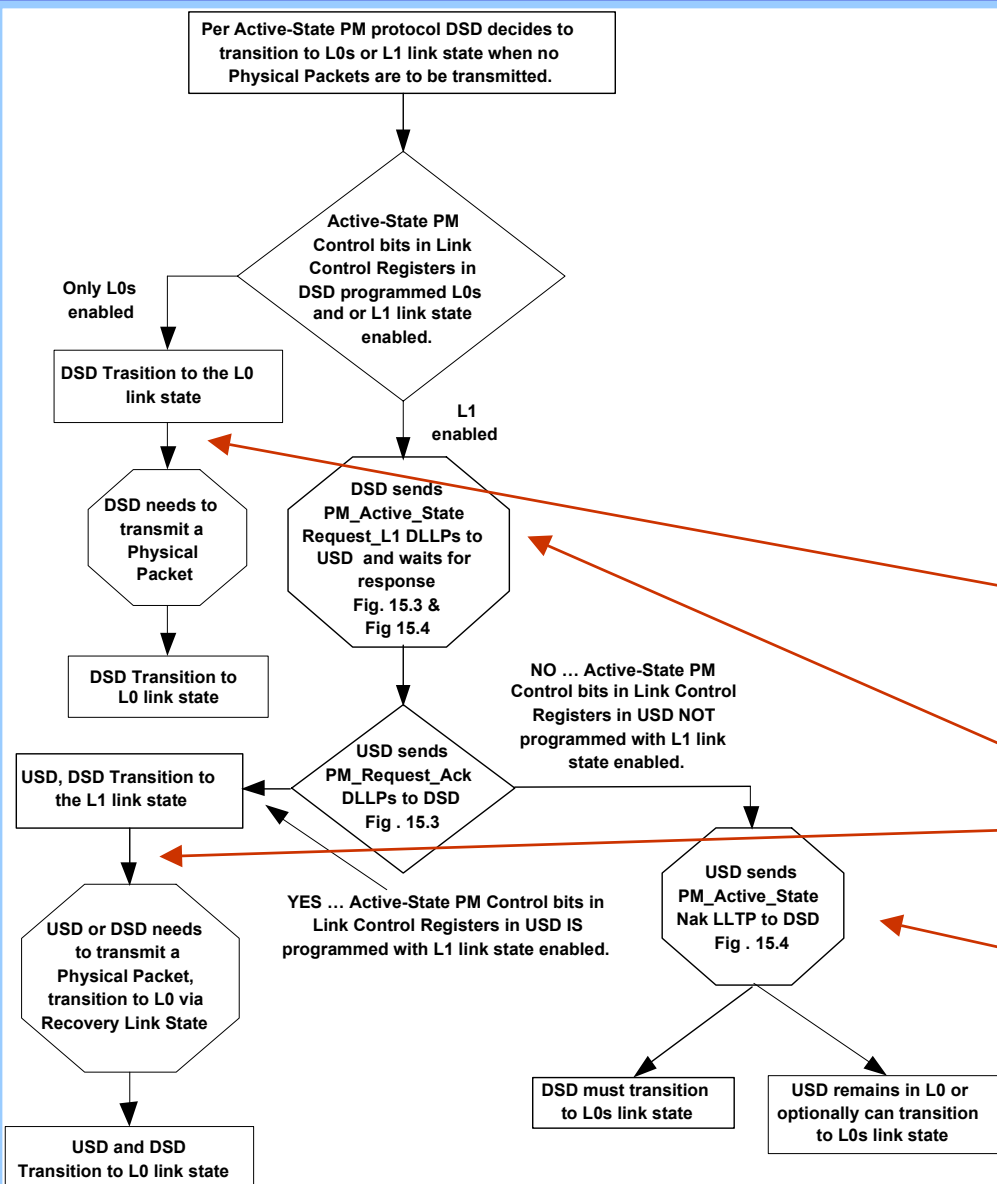
## Power Management Active-State Protocol

- The Active-State protocol transitions the PCI Express devices and associated link to the L0s or L1 link state.
  - The purpose of the L0s state is to permit the transmitters on any port to transition to a lower power link state when there are no Physical Packets to be transmitted. The transition to the L0s link state is independent for each port. Subsequently, the transition from the L0s to the L0 link state occurs independently on each port when a Physical Packet is ready to be transmitted.
  - The purpose of the L1 link state is to permit the transmitters on each end of a specific link to enter the L1 link state when there are no Physical Packets to be transmitted. Both PCI Express devices on the link must be in agreement to transition to the L1 link state. Otherwise, one will transition to the L0s link state (as a default) and the other will remain in the L0 link state or optionally transition to the L0s state.



## Power Management Active-State Protocol ... continued

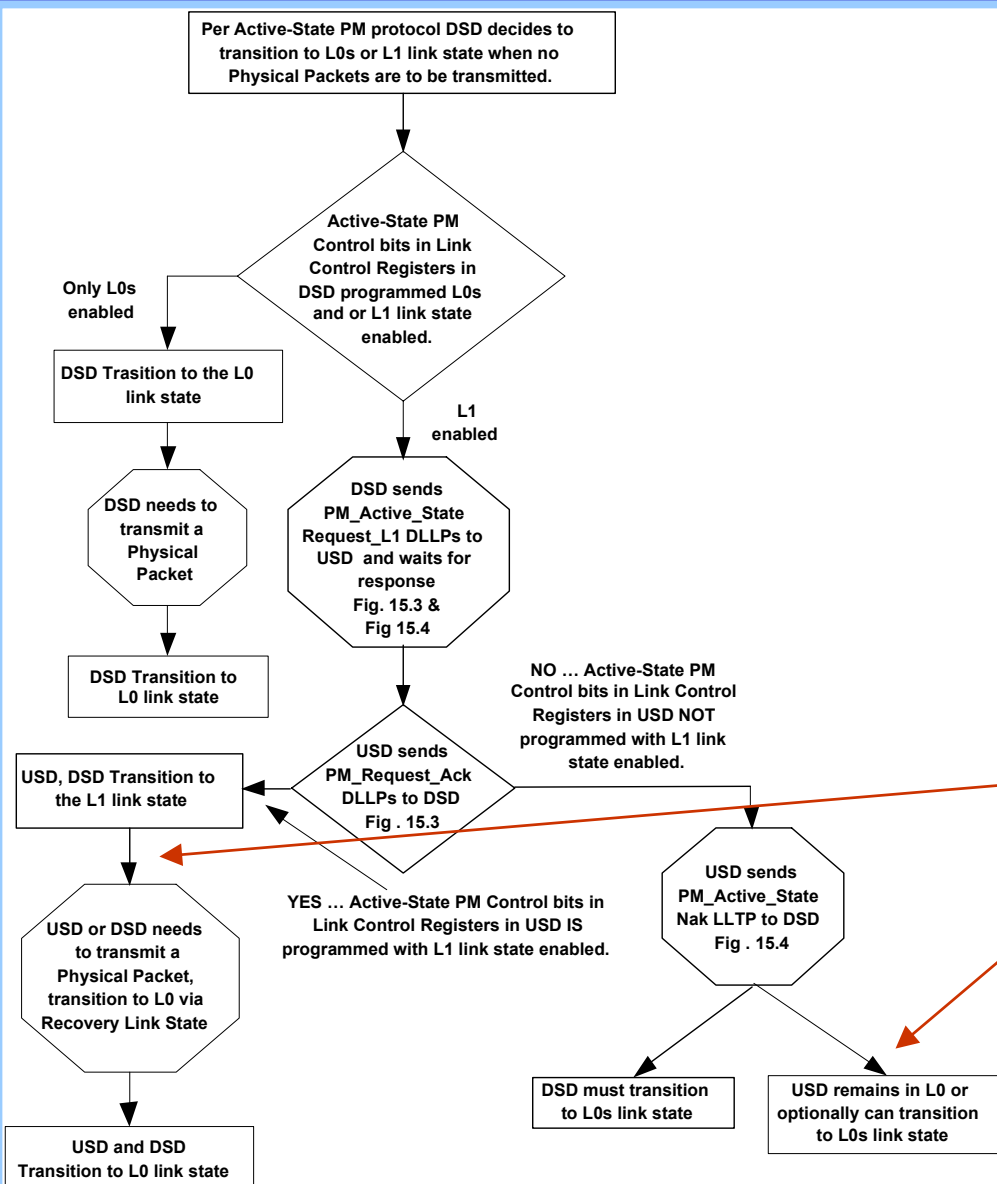
- The Active-State protocol ... continued
  - The determination to transition to the L1 link state is dependent on the value of the Active-State PM Control bits in the Link Control Registers programmed into the configuration registers blocks of the two PCI Express devices on the link. The selection is summarized in Table 15.9 in the Book.
  - The reason to select L0s versus L1 link state is reflective of the lower power of the L1 link state versus a longer latency to return to the L0 link state. The L0s link state is a compromise of not providing as much power reduction as L1, but a much quicker return to the L0 link state.



## Power Management Active-State Protocol ... continued

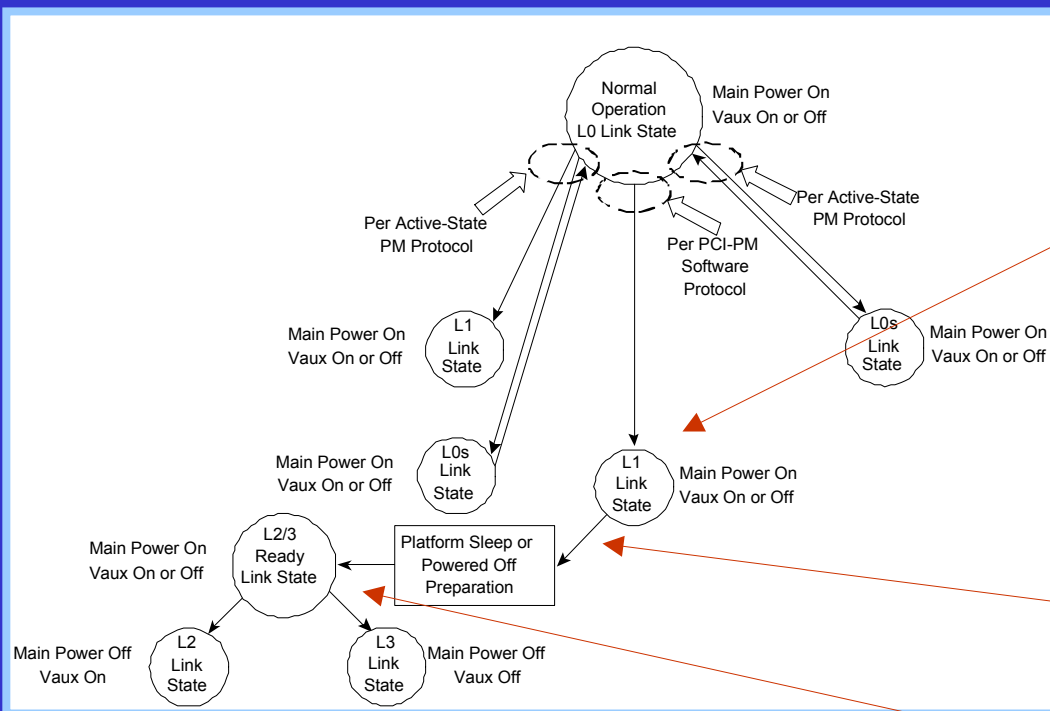
- Once the PCI Express device on the downstream side of the link (DSD) determines that no Physical Packets are ready to transmit, it requests a transition to the L0s or L1 link states. What it requests is dependent on what is enabled in the Active-State Control bits. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - If the Active-State Control bits only enable L0s link state, only the transmitters of the DSD transition to the L0s link state.
  - If the Active-State Control bits enable the L1 link state, the DSD transmits a PM\_Active\_State Request\_L1 DLLP and waits for PM\_Request\_Ack DLLP from the USD. If received from the USD the transition to the L1 link state by the USD and the DSD occurs. If received is the PM\_Active\_State Nack LLTP, the transition is to the L0s (if enabled) link state by the DSD occurs.





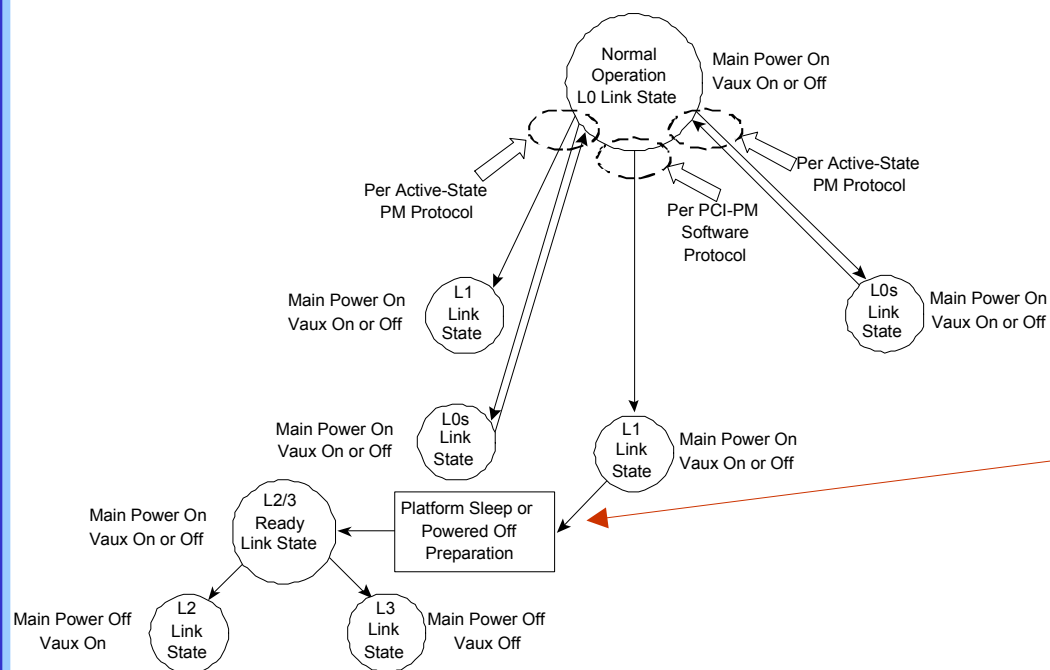
## Power Management Active-State Protocol ... continued

- The determination by the USD to respond with a PM\_Request\_Ack DLLP versus a PM\_Active\_State Nak LLTP is dependent on the value of Active-State Control bits in the USD. If L1 link state is enabled, a PM\_Request\_Ack DLLP will be transmitted in response to the receipt of a PM\_Active\_State Request\_L1 DLLP. If L1 link state is not enabled, a PM\_Active\_State Nak LLTP will be transmitted in response to the receipt of a PM\_Active\_State Request DLLP.
- If the USD transmitted a PM\_Request\_Ack DLLP, it transitions to the L1 link state. If the USD transmitted a PM\_Active\_State Nak LLTP, it remains in the L0 link state or if enabled by Active-State Control bits it transitions to the L0s link state.



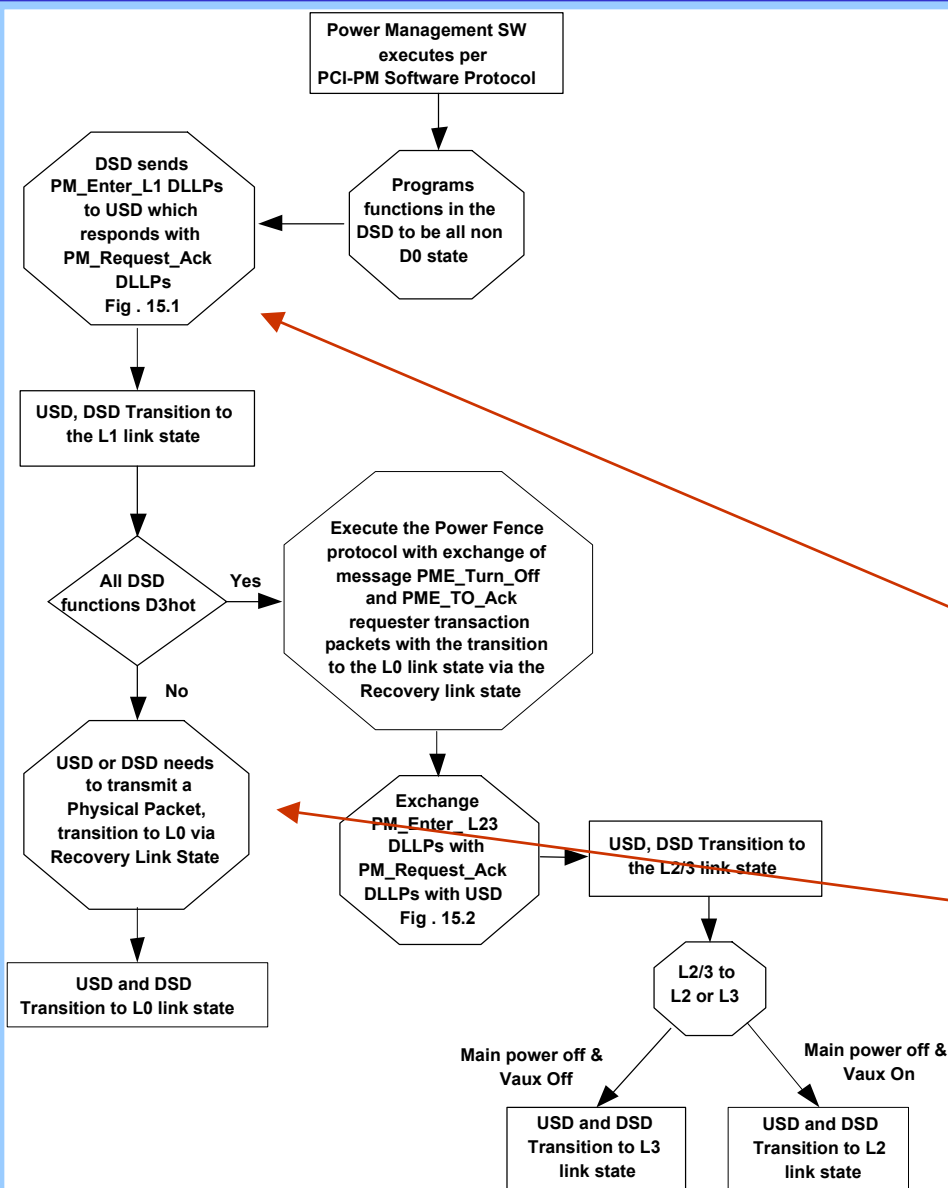
## Power Management PCI-PM Software Protocol

- The PCI-PM Software protocol supports the transition from the L0 to the L1 link states when all the functions in the PCI Express devices are programmed in non-D0 link states. The PCI-PM Software protocol permits the subsequent transition to the L2/3 Ready link state if all functions in the PCI Express devices are programmed into the D3hot state..
  - The transition from the L0 to the L1 link state requires exchanges of unique DLLPs.
  - Once in the L1 link state the transition to the L2/3 Ready link states requires the exchange of unique DLLPs and TLP.
  - Once in the L2/3 ready link state the transition to the L2 or L3 link state is dependent on the removal of the main power and/or Vaux.



## Power Management PCI-PM Software Protocol ... continued

- If the PCI-PM Software protocol had placed all of the functions of the PCI Express devices into the D3hot states, the Root Complex transmits a message PM\_TURN\_OFF requester transaction packet to prepare for the transition to the L2/3 Ready link state.
  - As part of the preparation the PCI Express device and associated link transitions to the L0 link state to transmit the message PM\_TO Requester transaction packet.
  - This is part of the Power Fence protocol discussed in detail in the Books.

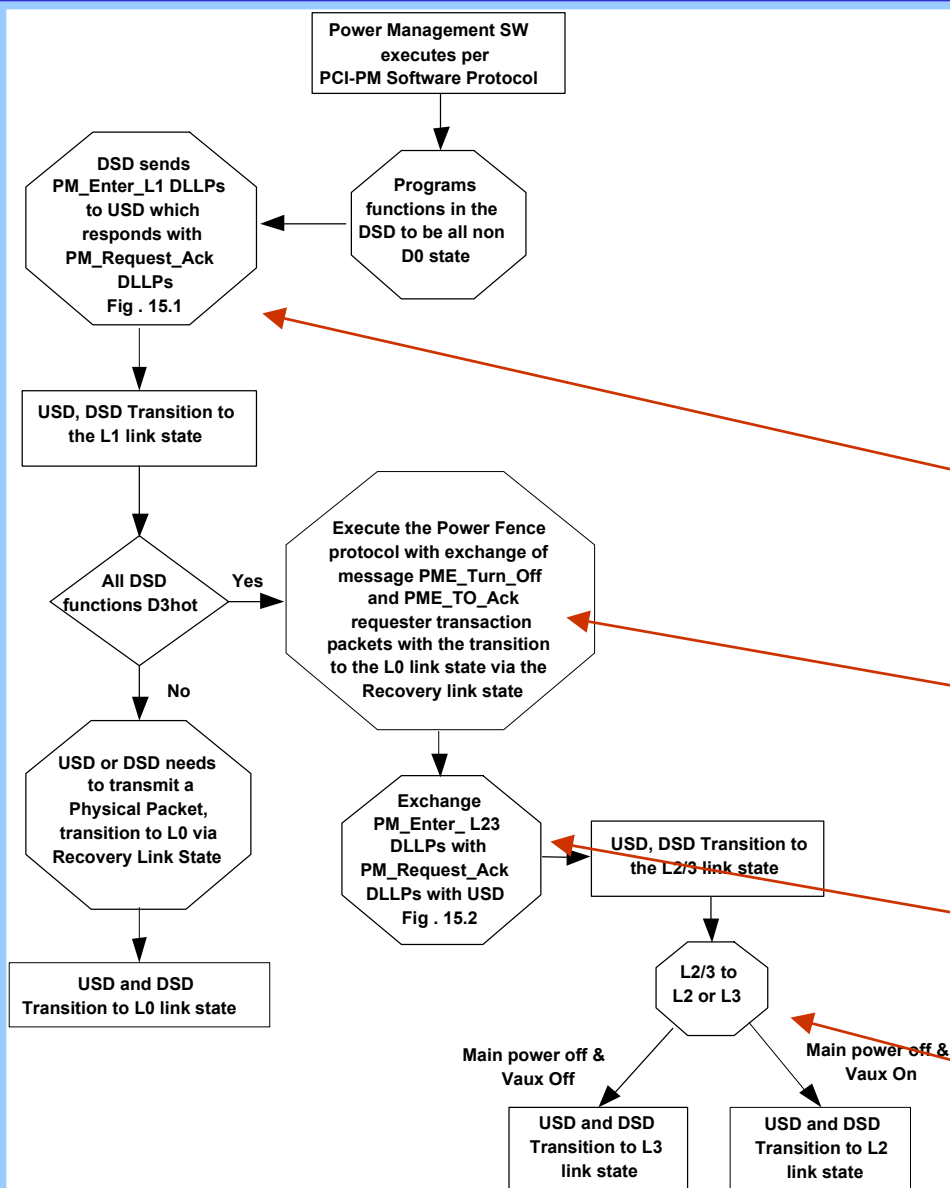


## Power Management PCI-PM Software Protocol ... continued

- Once the Power Management SW has determined to lower the power of PCI Express devices on a specific link, it programs the functions in the PCI Express device on the downstream side of the link (DSD).
- If all functions in the DSD are in non-D0 state but not all D3hot, the transition requested by the DSD is to L1 link state. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - Transition occurs with the exchange of the PM\_Enter\_L1 and PM\_Request\_Ack DLLPs.
  - The transition can only be initiated by the DSD with the transmission of the PM\_Enter\_L1 DLLP.
  - Transition from the L1 to L0 link state is via Recovery link state when Physical Packet must be transmitted

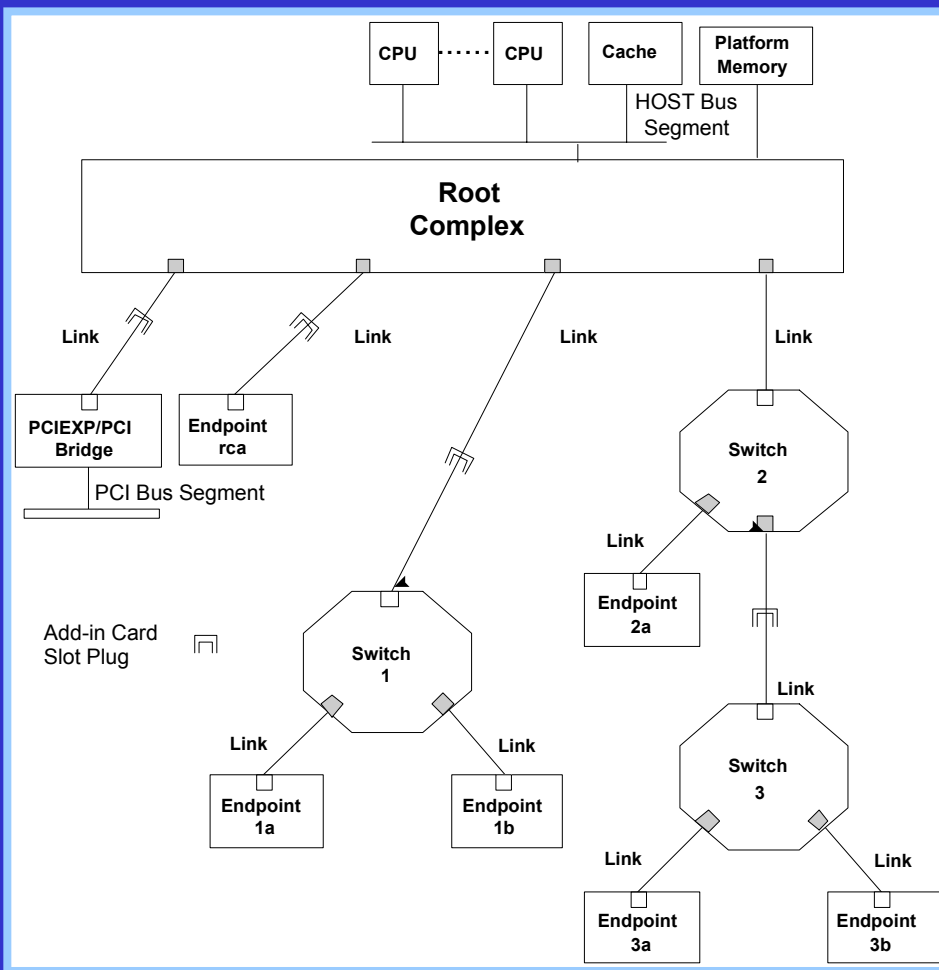
## Power Management PCI-PM Software Protocol ... continued

- If all functions in the DSD are programmed to the D3hot state, the transition requested by the DSD is for the L2/3 Ready link state executed in three parts. **Note: The Figure numbers referenced are the detailed flow charts in the Book.**
  - First, as discussed above, the two PCI Express devices and the associated link transition to the L1 link state with the exchange of the PM\_Enter\_L1 and PM\_Request\_Ack DLLPs.
  - Second, per the Power Fence protocol the message PME\_Turn\_Off and PME\_TO\_Ack requester transaction packets are exchanged after the transition from the L1 link state to the L0 link state via the Recovery link state.
  - Third, the DSD initiates the transition to the L2/3 Ready per the exchange of PM\_Enter\_L23 and PM\_Request\_Ack DLLPs.
  - When PCI Express devices and associated link are in the L2/3 Ready link state, the main power can be removed with or without the retention of Vaux.



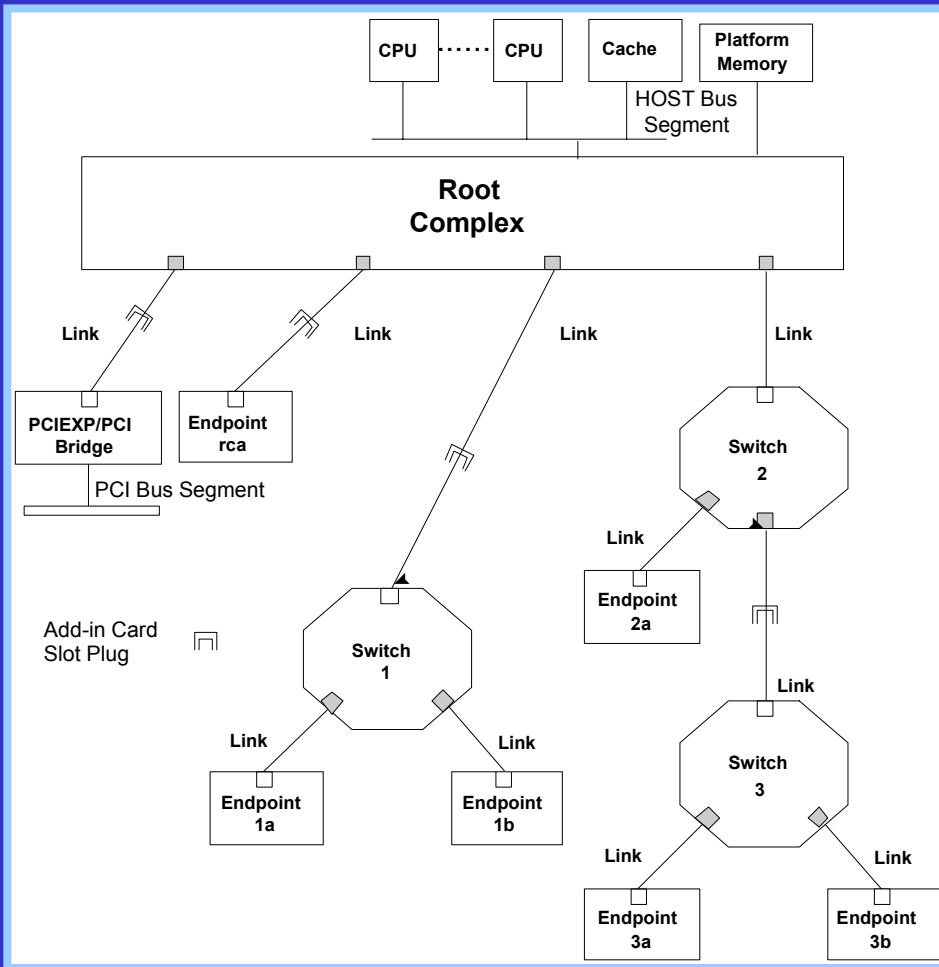
# Chapter 16

## Hot Plug Protocol



## Hot Plug Protocol Introduction

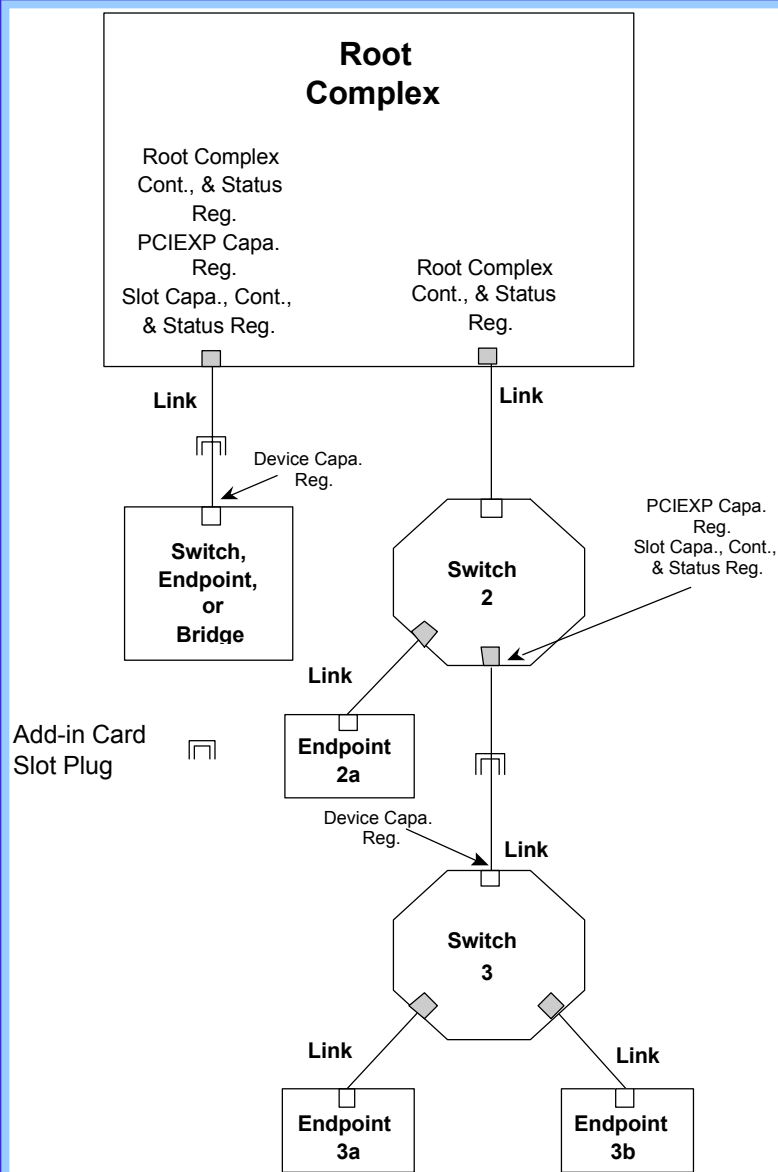
- An add-in card slot can optionally be defined supporting the Hot Plug protocol. The HP protocol permits add-in cards to be inserted and removed without powering off the entire platform.
- The slot compatible with the HP protocol can be attached directly to the downstream port of a Root Complex or a switch.
- The PCI Express device on the add-in card directly connected to slot may be an endpoint, switch, or bridge.



## Hot Plug Protocol Introduction ... continued

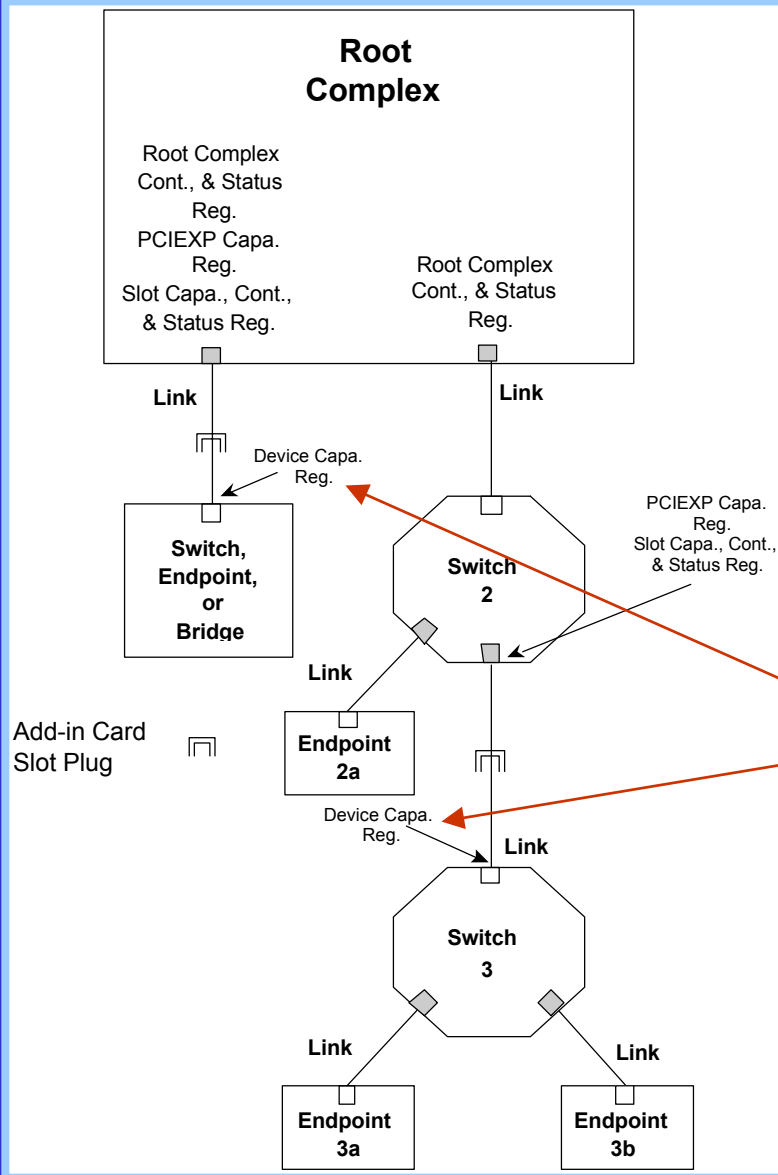
- In addition to the HP software there are three entities defined for the implementation of the HP protocol:
  - The chassis or add-in card has specific hardware for the HP protocol. The chassis is simply the portion of the platform associated with the HP compatible slot.
  - Specific registers are defined in the downstream ports of the Root Complex and switches. Specific registers are defined on the upstream port of the endpoints, switches, and bridges at the slot plug-in point on the add-in card. These registers are all within the configuration requester block of these PCI Express devices.
  - Several TLPs are defined for implementation by the HP protocol plus the use of interrupts associated with HP wakeup events.





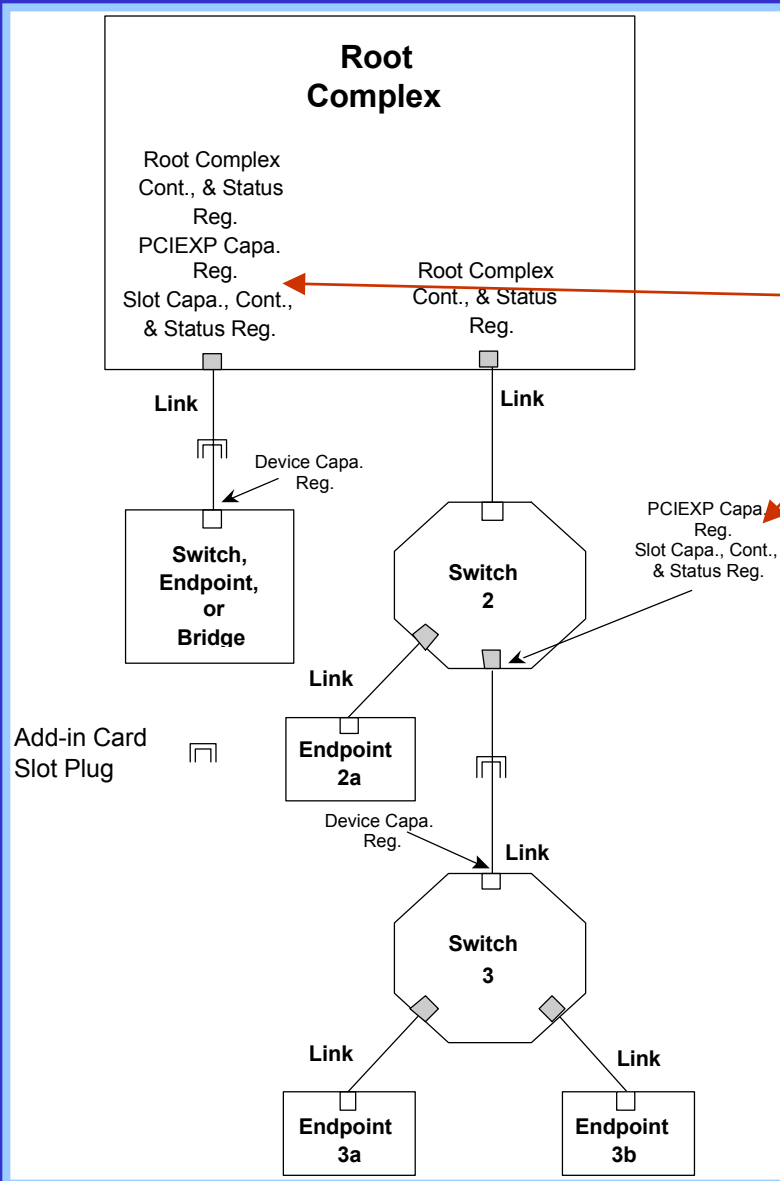
## Hot Plug Protocol Specific Hardware

- **Indicators:** There are two LEDs mounted either on the chassis or add-in card:
  - **Power Indicator:** Main power is on and applied to this slot and add-in card.
  - **Attention Indicator:** This is the add-in that needs service by the operator.
- **Attention Button:** This momentary button is pressed by the operator to request insertion or removal of an add-in card for the slot. It is mounted either on the chassis or add-in card.
- **Add-in Card present:** This is required but is implementation specific. It is a mechanism to indicate that an add-in card is inserted into the slot. Usually it is through the use of PRSNT# signal lines as defined by PCI.
- **Manually Operated Retention Latch (MRL):** Prevents insertion or removal of add-in card unless the mechanism is open. There is also an optional MRL sensor to indicate of the MRL is open or closed.
- **Power Fault:** A power controller specific to the slot can be optionally implemented. The power controller enables and disables power to the slot.



## Hot Plug Protocol Specific Registers

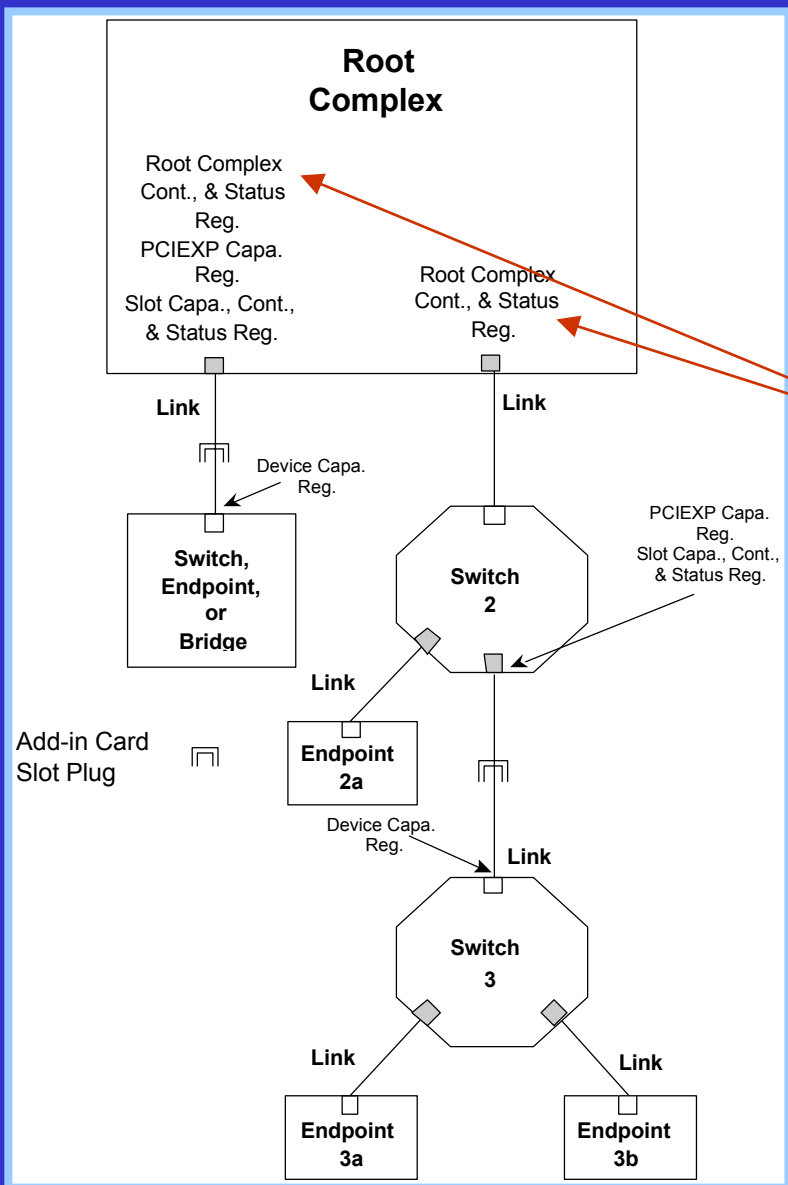
- Three groups for registers are defined for the HP protocol.
  - Registers within upstream port of the PCI Express device on the add-in card connected to the slot.
  - Registers within the downstream port of the PCI Express device on the platform connected to the slot.
  - Registers specific to Root Complex's downstream port associated with a hierarchy domain that contains a slot that supports the HP protocol.
- Registers within upstream port consists of the following:
  - **Device Capabilities Register:** Includes bits that define if the attention button, attention Indicator, or power indicator are implemented on add-in card.



## Hot Plug Protocol Specific Registers ... continued

Registers within the downstream port consists of the following:

- **PCIEXP Capabilities Register:** Includes a bit that indicates that the downstream port is connected to a slot.
- **Slot Capabilities Register:** Includes bits that define if the attention button, MSL sensor, attention Indicator, power indicator, or power controller are implemented on the chassis. It also contains other HP protocol qualifying bits.
- **Slot Control Register:** Controls the on/off of indicators and power controller. Also enables interrupt or wakeup event for HP hardware events discussed in later slide.
- **Slot Status Register:** Indicates if attention button pressed, MRL sensor or add-in card present changes state, power fault has occurred, or command is completed.



## Hot Plug Protocol Specific Registers ...continued

- Registers specific to Root Complex's downstream port consists of the following:
  - **Root Control Register:** Defines what type of interrupt is generated when a message PM\_PME requester transaction is received.
  - **Root Status Register:** This register provides the Requester ID (BUS#, DEVICE#, and FUNCTION#) of the PCI Express device that sourced the message PM\_PME requester transaction packet related to a wakeup event. It also indicates if any of these transactions are pending.

## Hot Plug Protocol with Normal Power

- Normal power is defined when PCI Express devices are operated with their functions in the D0 state.
- The Slot Capabilities and Device Capabilities Registers define the location if supported of HP related hardware: Attention button and Indicators on chassis or add-in card, and power fault detected, MSL sensor changed, or add-in card present change on chassis.
- There are five possible HP events: attention button pressed, power fault detected, MSL sensor changed, add-in card present change, or command completed.
  - Except for command completed, the other HP events are defined as HP hardware events. The Slot Status Register will indicate if any of these HP hardware events have occurred.
  - If an HP hardware event occurs, the value of the associated bit in the Root Control Register determines if a wakeup event or interrupt will occur at the Root Complex.
  - The HP event defined as command completed simply reflects the interaction with the HP software. When the HP software writes to the Slot Control Register to turn on/off/blink the attention indicator or power indicator, or turn power on/off via the power controller control a command occurs. To indicate completion of the command an interrupt will be sourced by the PCI Express device with the downstream port connected to the slot to alert the HP software via the Root Complex.
    - If the command is associated with the slot (i.e. hardware not on add-in card), the command is completed by the PCI Express device with the downstream port connected to the slot. If the command is associated with the add-in card (attention button or indicator on add-in card), part of the command completion is a message Hot-Plug requester transaction packet transmitted to the add-in card.

### Hot Plug Protocol with Normal Power ... continued

- Wakeup Event versus Interrupt
  - Per the above discussion, the PCI Express device with the downstream port connected to the slot a HP event will generate either a wakeup event or interrupt to indicate a HP hardware event has occurred. For the HP event related to command completion only an interrupt to the Root Complex is possible. If PCI Express device with the downstream port connected to the slot is at the Root Complex then a virtual internal interrupt occurs.
  - The wakeup event under normal power conditions consists of the PCI Express device with the downstream port connected to the slot sourcing a message PM\_PME requester transaction packet to the Root Complex. If PCI Express device with the downstream port connected to the slot is at the Root Complex then a virtual message PM\_PME requester transaction packet occurs.

### Hot Plug Protocol with Sleep or Lower Power Conditions

- Consider a PCI Express device in a low power condition with the associated functions are in D1, D2, or D3hot or in sleep. The occurrence of a HP hardware event (attention button pressed, power fault detected, MSL sensor changed, or add-in card present change) is possible. The occurrence of a HP event related to command completion is not defined for sleep or aforementioned low power conditions.
- The Slot Capabilities and Device Capabilities Registers define the location of HP related hardware
- Per the occurrence of the HP hardware event in low power conditions.
  - The Slot Status Register will indicate if any of these HP hardware events have occurred.
  - If a HP hardware event occurs, the value of the associated bit in the Slot Control Register will determine if a wakeup event occurs.
- Per the occurrence of the HP hardware event in sleep.
  - The Root Complex or PCI Express devices involved in the HP protocol is in sleep (main power off and VAUX retained) the occurrence of the HP hardware event can not be reported via the wakeup or interrupt. In the case of the attention button it is a not a HP hardware event to report because by definition of sleep the main power is off.
  - The PCI Express device with the downstream port connected to the slot, HP hardware event will execute the Wakeup protocol. Upon completion of the Wakeup protocol main power is on, this PCI Express device can source an interrupt or wakeup event.
  - If PCI Express device with the downstream port connected to the slot is at the Root Complex then a virtual internal interrupt or wakeup event occurs.
  - As stated in the previous slide the wakeup event is sourcing a message PM\_PME requester transaction packet to the Root Complex.

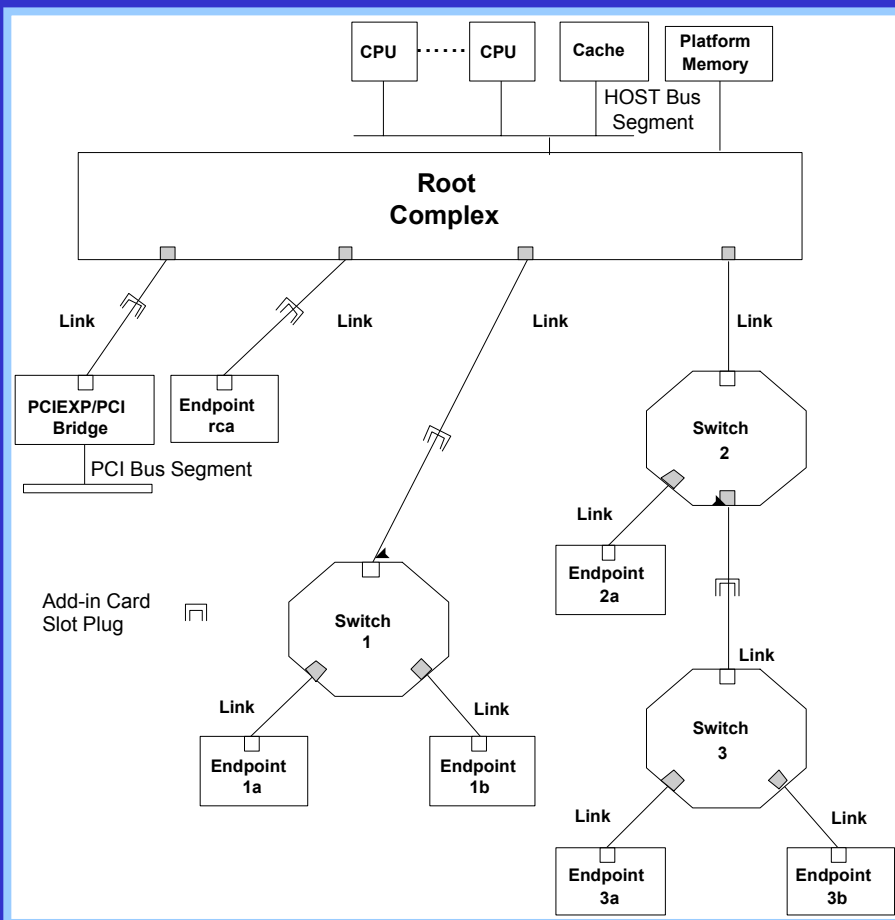
### Hot Plug Protocol and Transaction Layer Packets (TLPs)

- As mentioned in previous slide, the HP protocol implements message baseline requester transactions. These are divided into two groups: One group is for operation of the hardware. The other group is related to the reporting of a HP hardware event via wakeup event. Note, though not defined as a specific TLP related to the HP protocol, the execution of an interrupt to report a HP event can make use of message interrupt requester transaction packets.
- Operation of Hardware: The HP protocol defined message baseline requester transactions to execute the following:
  - Access attention indicator on add-in card to turn on, turn off, or blink the LED.
  - Access power indicator on add-in card to turn on, turn off, or blink the LED.
  - Report to PCI Express device with the downstream port connected to the slot that the attention button on add-in card was pressed.
- Support of wakeup event: The wakeup event simply uses the message PM\_PME requester transaction packet to alert the HP software via the Root Complex.



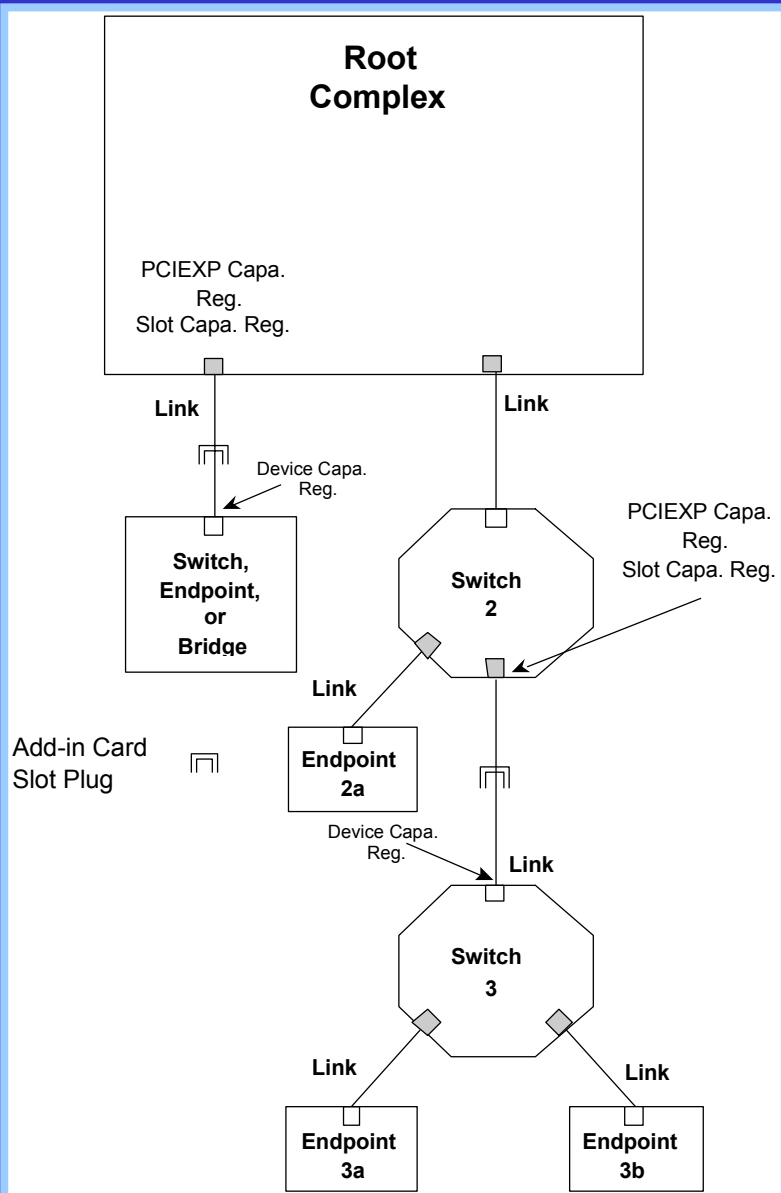
# Chapter 17

## Slot Power Protocol



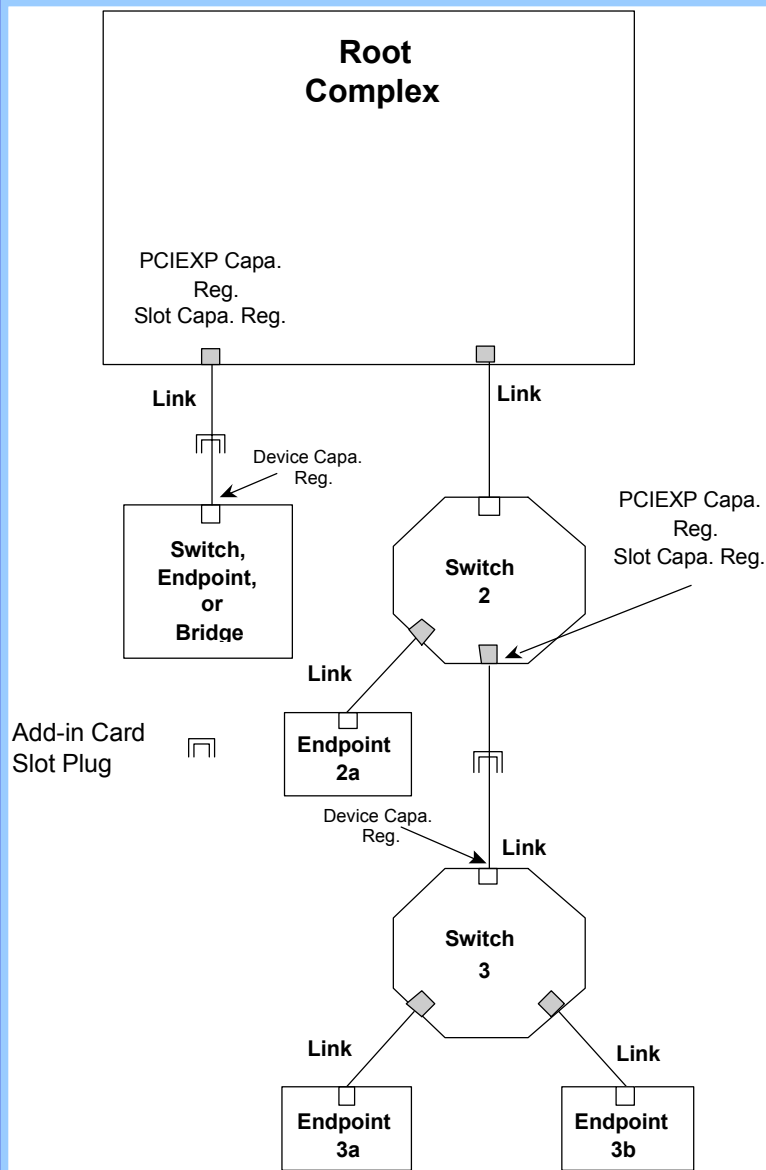
## Slot Power Protocol Introduction

- An add-in card must support the Slot Power (SP) protocol. The SP protocol permits Slot Power (SP) software to limit the power allocated and thus drawn by add-in cards via setting the current levels.
- The reason for power limitation to add-in cards primarily relates to cooling issues. It is also possible for SP software to manage the power allocated to all add-in cards in conjunction with the power requirements of PCI Express devices on the platform.
- The maximum amount of power that can be allocated to a add-in card is dependent on the size of the and the link width.
- The SP software can allocate less than the maximum power.



### Slot Power Protocol Introduction ... continued

- In addition to the SP software there are two entities defined for the implementation of the SP protocol:
  - Specific registers defined in the downstream ports of the Root Complex and switches. Specific registers defined on the upstream port of the endpoints, switches, and bridges at the slot plug point on the add-in card. These registers are all within the configuration requester block of these PCI Express devices.
  - A specific TLP is defined for implementation by the SP protocol.



## SP Protocol Specific Registers

- Two groups for registers are defined for the SP protocol.
  - Registers within upstream port of the PCI Express device on the add-in card connected to the slot.
  - Registers within the downstream port of the PCI Express device on the platform connected to the slot.
- Registers within upstream port consists of the following:
  - Device Capabilities Register:** This register contains the Captured Slot Power Limit Value a binary value that provides a base number for the maximum wattage to be used by the add-in card. It also includes Captured Slot Power Limit Scale bits that define the multiplier of this number.
- Registers within the downstream port consists of the following:
  - PCIEXP Capabilities Register:** Includes a bit that indicates that the downstream port is connected to a slot.
  - Slot Capabilities Register:** This register contains the Slot Power Limit Value, a binary value that provides a base number for the wattage to be allocated to the slot. It also includes the Slot Power Limit Scale bits that define the multiplier of this number.

### SP Protocol, Registers, and TLP

- As mentioned in previous slide, the SP protocol defines two sets of bits in the downstream port of the Root Complex or switch connected to the slot: Slot Power Limit Value and Slot Power Limit Scale. A similar two sets of bits in the upstream port of the endpoint, switch, or bridge connected to the slot are defined: Captured Slot Power Limit Value and Captured Slot Power Limit Scale.
- The combination of the Slot Power Limit Value and Slot Power Limit Scale bits defined the maximum wattage allocated to the slot. When data is written to either of these bit groups by the SP software, the value must be updated in the Captured Slot Power Limit Value and Captured Slot Power Limit Scale bits. The Captured Slot Power Limit Value and Captured Slot Power Limit Scale bits define the maximum wattage that the add-in card can use.
- Whenever the SP software writes to the Slot Power Limit Value or Slot Power Limit Scale bits, a message Set\_Slot\_Power\_Limit requester transaction packet is transmitted downstream to update the to the Captured Slot Power Limit Value or Captured Slot Power Limit Scale bits.
- Until the Captured Slot Power Limit Value or Captured Slot Power Limit Scale bits are first updated by the aforementioned TLP, there is a maximum default wattage that can be consumed by the add-in card.
- The message Set\_Slot\_Power\_Limit requester transaction packets are also transmitted to update the Captured Slot Power Limit Value or Captured Slot Power Limit Scale bits when the DLCMSM in the PCI Express device on the downstream port indicates a transition from Link\_DOWN to Link\_UP. The DLCMSM (Data Link Control and Management State Machine ) is in the Data Link Layer.

# The Complete PCI Express Reference Topic Group 5 Tutorial

Disclaimer: Intel, Research Tech Inc. and the authors of this tutorial make no warranty for the accuracy or use of the information. No direct or indirect liability is assumed and the right to change any information without notice is retained.

## Design Tools for PCI Express

The PCI Express specification is not organized by clear design topics, does not provide sufficient details to easily master PCI Express, and does not provide in depth illustrations and flowcharts to assist in designing components. Intel recognized that simply summarizing or re-wording the specification as typically done in the other design tools in the industry is insufficient ... more extensive information is are needed.

To provide designers with PCI Express design and implementation information that is easy to master and detailed enough to assist in correct design, two design tools are available:

Six Detailed Tutorials and a new and exhaustively detailed design book.

These design tools focus on Six Quick and Easy Topic Groups which simplify the mastery of PCI Express. They save a designer weeks of trying to unravel the specification and provide the assurance of correct design implementation the first time.

## Design Tools for PCI Express

### The “Book”

The primary design tool that provides total design mastery is *The Complete PCI Express Reference* book written by Edward Solari and Brad Congdon and published by Intel ... referred to as the “Book”.

The Book provides the complete and extensive narrative of detailed figures (over 250), detailed design flow charts, and exhaustive tables for the complete understanding and design assistance in over 1000 pages. The Book can be ordered at [www.amazon.com](http://www.amazon.com) ... ISBN # 0971786194.

### Detailed Tutorials

Six free Detailed Tutorials ... One self paced tutorial for each of the Six Quick and Easy Topic Groups. Each introduces PCI Express information with a narrative that complements detailed figures, flow charts, and tables for each specific Topic Group from the Book. The six free Detailed Tutorials are available at [www.intel.com/intelpress/pciexpresscomplete](http://www.intel.com/intelpress/pciexpresscomplete).

This Detailed Tutorial is of Topic Group 5  
Detailed Tutorial: *Other Hardware Topics*  
References in the Book: *Chapters 18 to 21*



## PCI Express in Six Topic Groups

### Topic Group 1

**Tutgroup1: *Platform Architecture and Accessing of Resources within Architecture***

**References in the Book: *Chapters 1 to 4***

### Topic Group 2

**Tutgroup2 : *Packets' and Layers' Specifics and Errors***

**References in the Book: *Chapters 5 to 9***

### Topic Group 3

**Tutgroup3 : *Transaction Ordering and Flow Control Part 1 and 2 Protocols***

**References in the Book: *Chapters 10 to 12***

### Topic Group 4

**Tutgroup4 : *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

**References in the Book: *Chapters 13 to 17***

### Topic Group 5

**Tutgroup5 : *Other Hardware Topics***

**References in the Book: *Chapters 18 to 21***

### Topic Group 6

**Tutgroup6 : *Software Considerations***

**References in the Book: *Chapters 22 to 24***

# Other Hardware Topics

## Chapters 18 to 21

### Topic Group 5

The PCI Express specification includes features that support PCI compatible interrupts and lock function to retain PCI software compatibility. The PCI Express specification also defines a set of add-in card sizes that are mechanically compatible with a PCI chassis. Independent of PCI are the electrical attributes that are specific to PCI Express's need to transmit across a differently driven signal lines. To complete the discussion, advanced switching is summarized.

**Summary:** The PCI Express platform retains software compatibility with PCI software through the use of virtual PCI devices and configuration address space. An extension of the compatibility to PCI software is the implementation of PCI compatible interrupts and the lock function. The hardware INTx signal lines are implemented via emulation by message requester transactions. The hardware LOCK# signal line of PCI is implemented via emulation by memory requester transactions and a message requester transaction to terminate the Lock function.

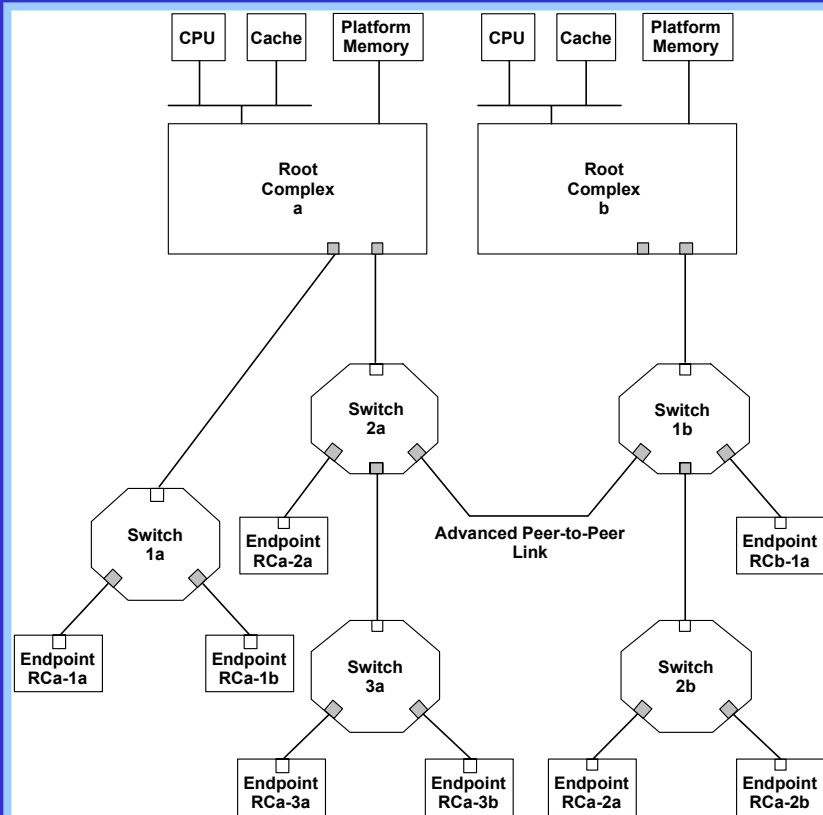
In order for the not require a re-design of a non-mobile or a mobile PCI chassis, PCI Express adopts the basic add-in card size of PCI.

PCI consists of multiple signal lines referenced to a clock signal line. In order to minimize the signal line between PCI Express devices a pair of differently driven signal lines are defined with an integrated reference clock.

Advanced switching is a place holder for future message protocol between two PCI Express fabrics. signal line are

# Chapter 18

## Advanced Switching

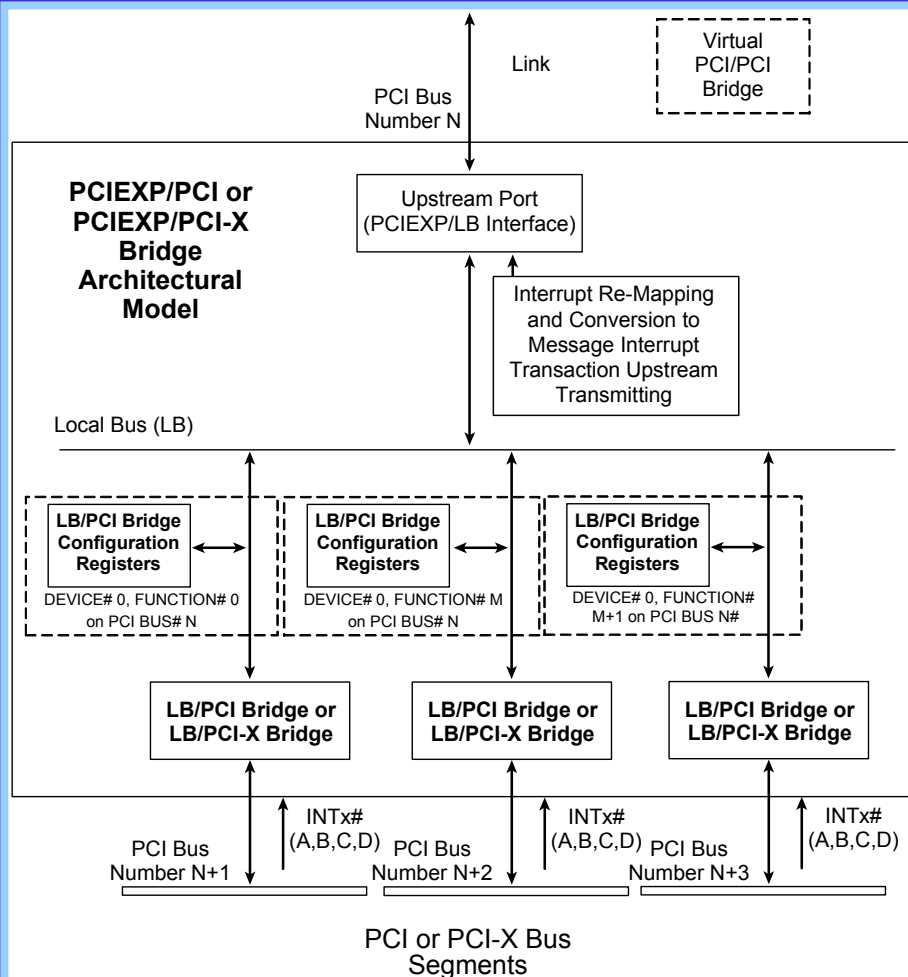


## Advanced Switching

- The Advance Switching protocol was in the original PCI Express specification but was subsequently removed. The “Advanced PCI Express Packet Switching Specification” should be available by 2004.
- For completeness some basics of this protocol is as follows. (These may change as the specification is refined. They are only here for background information):
  - Purpose is to provide communication between two or more PCI Express fabrics. As illustrated, the two PCI Express fabrics are defined by Root Complex a and its downstream devices and Root Complex b and its downstream devices.
  - The interconnection between the fabrics is the advanced peer-to-peer link which follows the protocols of a regular link and connected to only switches’ downstream ports .
  - The communication mechanism is message advance switching requester transaction packets.
  - The addressing mechanism used is a 15-bit global address space defined as the Route Identifier (RID)
  - The concept is that a wide range of formats can be contained within the Data field of a message requester transaction packet. This permits software supporting different communication protocols.

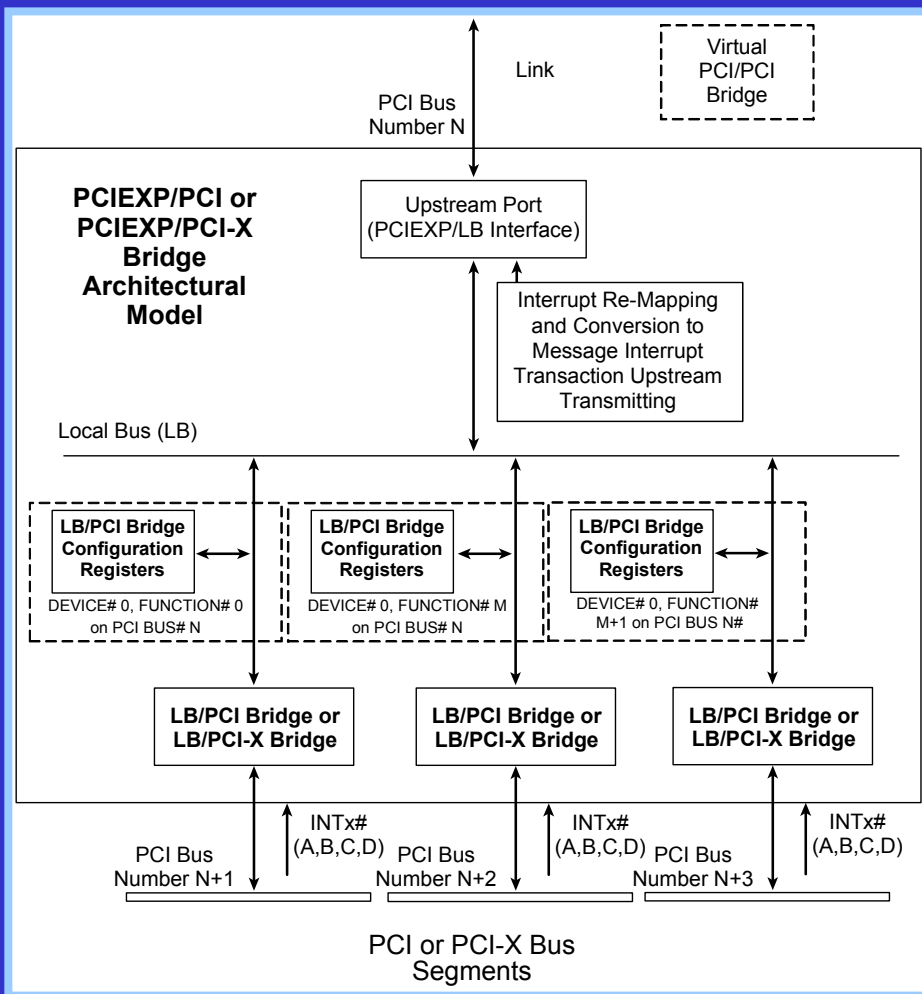
# Chapter 19

## Interrupts



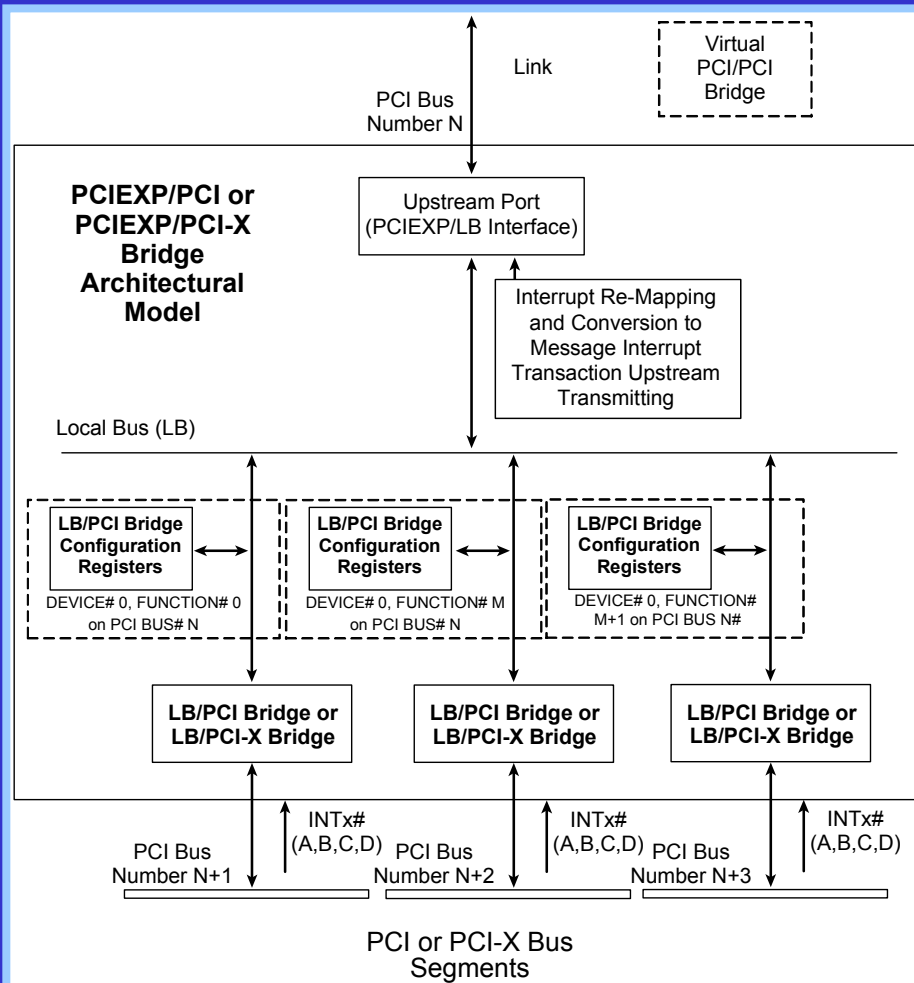
## Interrupts

- Legacy endpoints and PCI devices downstream of a bridge define MSI (Memory Signaled Interrupts) and discrete interrupt signal lines as the methods to request interrupt service.
- PCI Express endpoints only define MSI. MSI uses the memory write transaction protocol across the links.
- There are no additional signal lines on the links to support discrete interrupt signal lines from the legacy endpoints or from the bridges.
- The legacy endpoints “internally” may support discrete interrupt signal lines, but must convert to “some other mechanism” at the PCI Express port. Bridges must support discrete interrupt signal lines on the PCI bus segment, but must also convert to “some other mechanism” at the PCI Express port.
- The “some other mechanism” is the conversion of discrete interrupt signal lines into message interrupt requester transaction packets. These packets only flow upstream to the Root Complex to emulate discrete interrupt signal line assertion and deassertion. The interrupt is acknowledged by HOST bus segment software accessing the PCI or PCI Express device from the via the Root Complex.
- Bridges will be used to exemplify this conversion for both bridges and legacy endpoints.



## Interrupts ... continued

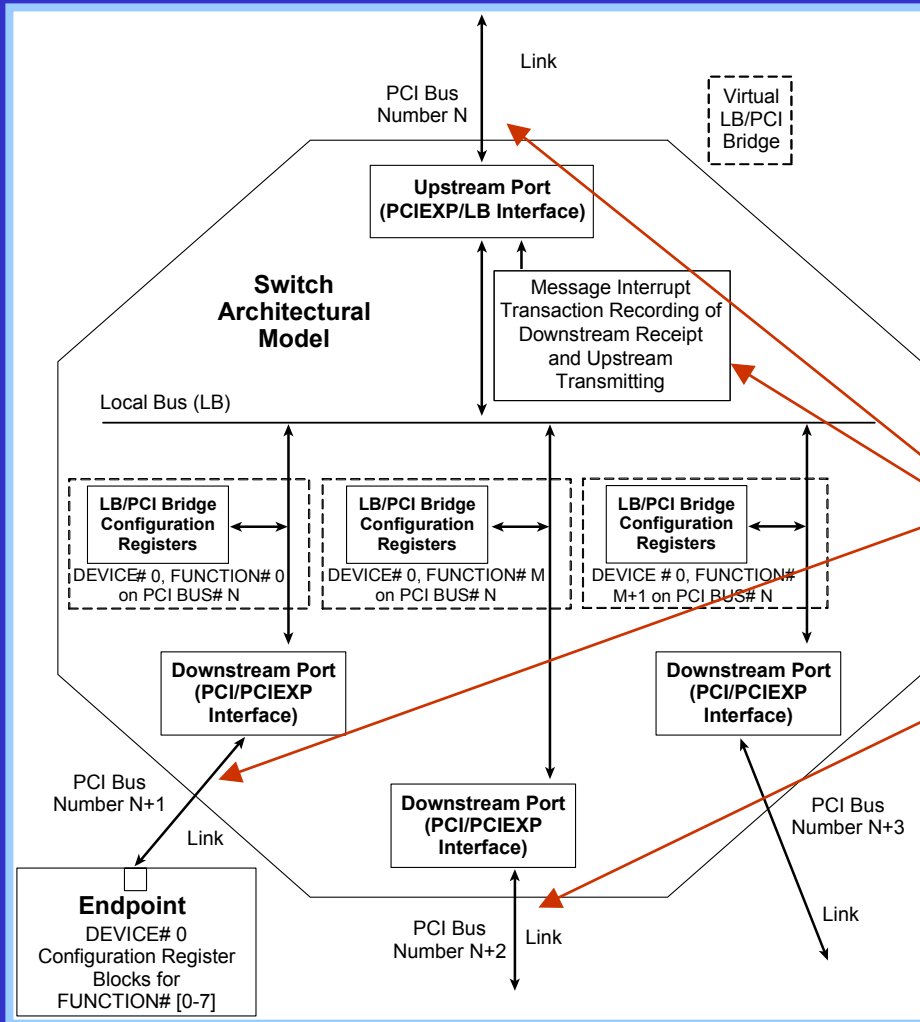
- The assertion of a discrete interrupt signal lines is emulated by the message Assert\_INTx requester transaction packet.
- The deassertion of a discrete interrupt signal lines is emulated by the message Dessert\_INTx requester transaction packet.
- Four discrete interrupt signal lines are defined (A to D). Thus four pairs of message Assert\_INTx and message Deassert\_INTx (x = A to D) requester transaction packets are defined
- Each message interrupt requester transaction pair operate independently of each other, just like the four discrete interrupt signal lines operate independently.
- As detailed in the Book there is interrupt remapping as they flow through a bridge.



## Interrupts ... continued

- Bridges and endpoints must keep track of the discrete signal lines that have shared interrupts.
  - For example, the “or” function (open collector) implementation of a shared interrupt signal line “A” means the first interrupt asserts the interrupt signal line and the second interrupt also asserts the interrupt signal line but is not “seen”. Once the first interrupt is serviced and the continued assertion of the interrupt signal line indicates that the second interrupt has occurred and had not been serviced.
  - The bridge must emulate the “or” function of the shared interrupt signal lines by transmitting upstream the message Assert\_INTA requester transaction packet for the first interrupt, but not transmitting one for the second. The occurrence of the second is not “seen” by the bridge because of the “or” function of the discrete interrupt signal line “A”. Once the second interrupt is serviced the discrete interrupt signal line is deasserted and the bridge transmits upstream a message Deassert\_INTA requester transaction packet.





## Interrupts ... continued

- The switches port all message interrupt requester transaction packets received on the downstream port upstream.
- Each switch must independently keep track of the message interrupt requester transaction packets received at each downstream port and independently keep track of the packets for "A" versus "B", etc. Effectively the switch must merge the message interrupt requester transaction packets to emulate the "or" function.
- For example, in a mechanism similar to the bridge's the receipt of the "first" message Assert\_INTA requester transaction packet at one downstream port is recorded and transmitted upstream. The second message Assert\_INTA requester transaction packet received at another port is also recorded but is not transmitted upstream.
- Once a message Deassert\_INTA requester transaction packet is received on BOTH downstream ports, the switch will transmit upstream one message Deassert\_INTA requester transaction packet.
- The above protocol is executed independently for each interrupt (A to D) across all the downstream ports.
- See the Book for more detailed discussions.

# Chapter 20

## Lock

## LOCK

- The Lock Function is simply the implementation of Exclusive Hardware Access. That is, a PCI Express device can only be accessed by the Root Complex and no other legacy endpoints. In the case of a PCI device downstream of a bridge, only Root Complex or other PCI devices as bus masters downstream of the bridge that are currently participating in the Lock Function.
- Legacy endpoints downstream of the Root Complex can become a locked legacy endpoint and participant in the Lock Function. PCI devices downstream of a bridge can become a locked target and participant in the Lock function via the bridge.
- PCI Express endpoints can never become participate in the Lock Function.
- It is also possible to have Exclusive Software Access to a PCI Express device or a PCI device, this is implementation specific.
- The purpose of the Lock Function is to implement exclusive access by the Root Complex to prevent data changes during the course of software on the HOST bus segment being executed. For example, for the use of semaphores.

### LOCK ... continued

- The Lock Function is implemented as follows:
  - A memory read requester transaction packet with the LOCK bit asserted in the Header field is sourced from the Root Complex to downstream PCI Express device. If the PCI Express device can participate it sources a successful completer transaction packet with LOCK bit asserted in the Header field.
  - With the successful receipt of the completer transaction packet at the Root Complex, the Lock Function has been established between the Root Complex and the PCI Express device. Consequently, the Root Complex is the only device that can access the PCI Express device. Each downstream port of the Root Complex can establish the Lock Function independently and in parallel.
    - If the PCI Express device is connected directly to the Root Complex the access is only from the Root Complex that is participating in the Lock Function.
    - If the PCI Express device is connected to a switch, it the switch's responsibility not to forward a requester transaction packet unless it is sourced from the Root Complex as defined by the Requester ID in the Header field.
  - To terminate the Lock function the Root Complex transmits downstream a message unlock requester transaction packet.
- See the Book for more detailed discussions.

# Chapter 21

## Mechanical and Electrical Overview

## Mechanical ... Add-in Card Sizes

- The primary PCI Express mechanical specifications are *PCI Express Card Electromechanical* and *Mini PCI Express Card Electromechanical*. The Book and this tutorial does not try to replace these mechanical drawings provided in these specifications. The mechanical information provided in these specifications are straight forward and does not need further clarification. The pin assignments for the connectors do require some commentary which is provided in subsequent slides.
- PCI Express defines two basic form factors: non-mobile and mobile. There are other form factors being considered as discussed in the next slide.
  - **Non-mobile:** The basic add-in cards used by PCs and servers. The non-mobile form factors are based on PCI to retain compatibility to the PCI chassis form factor. The add-in card sizes defined are short length, standard length, and low profile.
    - The number of lanes currently defined are x1, x2, x4, and x16. Not all number of lanes are defined for all possible add-in card sizes.
    - The Book details the combination of lanes numbers relative to connectors versus the lanes implemented by the different add-in card sizes.
    - 12, 3.3, and 3.3 aux (Vaux) volts are defined.
  - **Mobile:** The basic add-in cards used by battery operated PCI Express platforms and is as the Mini PCI Express card. There is currently one mobile form factor are based indirectly on the Mini PCI Card Type III. Essentially, a Mini PCI Express Card is the same length as Mini PCI Card Type III but one half the width. Thus, two Mini PCI Express Cards fit into the space of one Mini PCI Card Type III.
    - The number lanes is currently fixed as X1, but pins are reserved to allow an future possible x2 implementation.
    - 1.5, 3.3, and 3.3 aux (Vaux) volts are defined.

### Mechanical ... Add-in Card Sizes ... continued

- **Other Form Factors:** Different organizations are at different levels of proposing different packaging schemes. Each of these proposals have varying degrees in terms of completeness of mechanical information. The current list of other form factors are as follows:
  - **NEWCARD:** Defined by the PCMICA group as the replacement for Cardbus. It implements either a PCI Express x1 interface among others. Two card widths are defined with a fixed card length. NEWCARD is defined for both mobile and non-mobile applications.
  - **Sever I/O Modules:** Currently this is a preliminary sever centric proposal that defines four form factors of self contained modules. The modules are fixed length with either a base or full height. Each of the two heights are defined as single wide and double wide.
  - **Advance TCAs:** Defined for use in carrier grade communications equipment and PCI Express and other interfaces. Two card sizes are defined.
  - **Cable Modules:** Minimally defined at this time. Its purpose is enterprise class systems, it implements a PCI Express interface, and there is power distribution through the cable.

## Mechanical ... Non-Mobile Connector Pins

- **REFCLK** (Input to add-in card): Similar to the CLK signal line of PCI and PCI-X except it is used to provide the basis for a reference clock integrated into the address/data stream per 8/10b encoding. Consists of two differentially driven signal lines: REFCLK+ and REFCLK-.
- **PETpx and PETnx**: LANE# Differential Pair Transmitter (Output from add-in card): The signal lines that implement the PCI Express lane of the link. Consists of two differentially driven signal lines: positive (p) and negative (n). One pair defined for each lane.
- **PERpx and PER nx**: LANE# Differential Pair Receiver (Input to add-in card): The signal lines that implement the PCI Express lane of the link. Consists of two differentially driven signal lines: positive (p) and negative (n). One pair defined for each lane.
- **PERST#** (Input to add-in card): Signal line is used as a reference point for Cold Reset and Warm Reset, It is also an indicator for the power level of main power.
- **JTAG** (Optional): Five connections used for platform or non-mobile add-in card production testing. JTAG is defined by IEEE Standard 1149.1.
  - TRST# (Input to add-in card): Test reset.
  - TCK (Input to add-in card): Test clock.
  - TDI (Input to add-in card): Test data in.
  - TDO (Output to add-in card): Test data out.
  - TMS (Input to add-in card): Test mode select.



### Mechanical ... Non-Mobile Connector Pins ... continued

- **SMBus** (Optional): System Management bus (SMBus) can be implemented between PCI, PCI-X, and PCI Express components on the platform and the add-in cards for a low power and lower bandwidth serial bus. The bus consists of SMBCLK (input to the add-in card) and SMDAT (input/output). This general purpose bus is used for purposes that range from platform management to inputs for equipment.
- **PRSNT1#** (Input to add-in card) and **PRSNT2#** (Output from add-in card): Originally defined by PCI and PCI-X as add-in card presence detect, these signal lines are defined by PCI Express for the Hot Plug protocol.
- **WAKE#** (Output from add-in card): Required for add-in cards and platforms that implement the Wakeup protocol. This signal line is used in conjunction with Vaux as part of the Wakeup Protocol.
- **Power Pins** (Input to add-in card): +12, +3.3, and 3.3 Vaux volts. 3.3 Vaux is also known simply as Vaux.
- **Reserved**: These signal lines can only be redefined by the PCI-SIG. Neither platforms nor add-in cards can connect anything to these signal lines.
- **GND**: These signal lines are connected directly to the ground plane of the platform.

### Mechanical ... Mobile Connector Pins

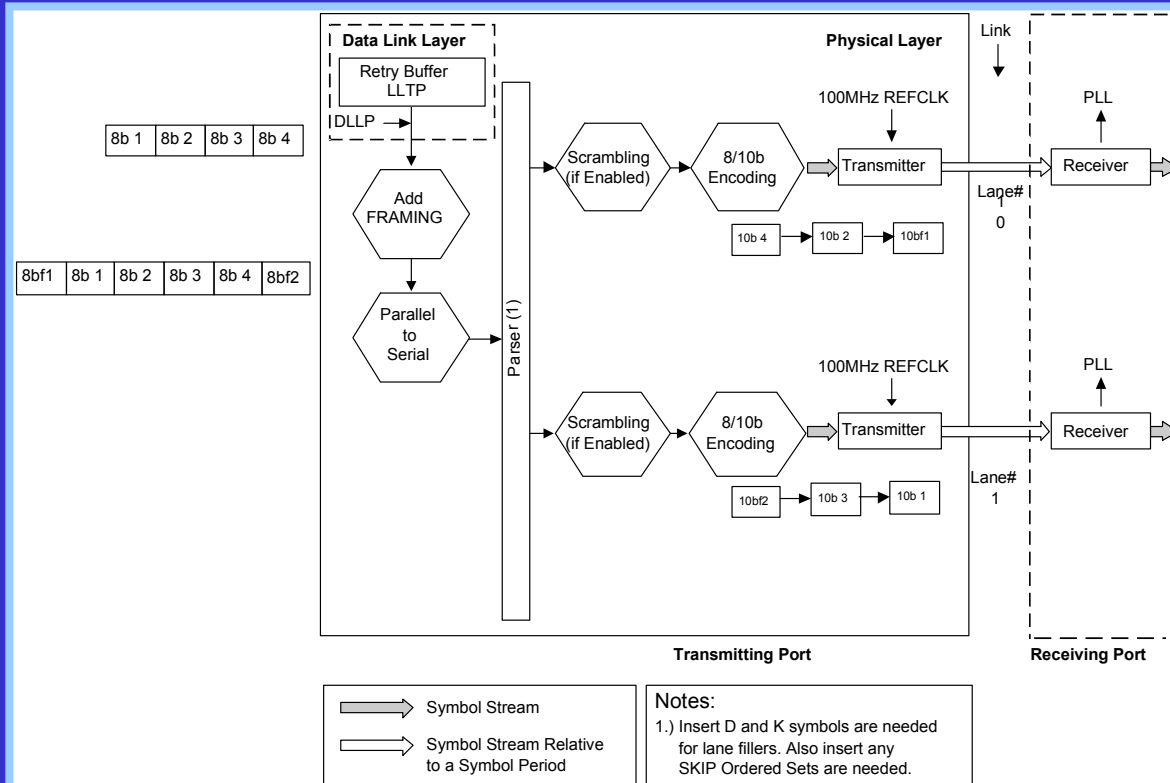
- **REFCLK** (Input to add-in card): Similar to the CLK signal line of PCI and PCI-X except it is used to provide the basis for a reference clock integrated into the address/data stream per 8/10b encoding. Consists of two differentially driven signal lines: REFCLK+ and REFCLK-.
- **PETpx** and **PETnx**: LANE# Differential Pair Transmitter (Output from add-in card): The signal lines that implement the PCI Express lane of the link. Consists of two differentially driven signal lines: positive (p) and negative (n). One pair defined for each lane.
- **PERpx** and **PERnx**: LANE# Differential Pair Receiver (Input to add-in card): The signal lines that implement the PCI Express lane of the link. Consists of two differentially driven signal lines: positive (p) and negative (n). One pair defined for each lane.
- **PERST#** (Input to add-in card): Signal line is used as a reference point for Cold Reset and Warm Reset, It is also an indicator for the power level of main power.
- **CLKREQ#** (Output from add-in card ... mobile add-in cards only): For mobile add-in cards that implement the PCI Express the assertion (active low) of this signal lines enables the REFCLK to be sourced from the platform to the add-in card. This is an open collector signal line.
- **PERST#** (Input to add-in card): Typically the PERST# signal line is used as a reference point for Cold Reset as an indicator for the power level of main power. It is also used as reference point for a Warm Reset. See Chapter 13 for more information.

### Mechanical ... Mobile Connector Pins ... continued

- **USB** (Optional): Universal Serial Bus. Defined by the USB Specification revision 2.0. The bus consists of differentially driven pair USB\_D+ and USB\_D- (input/output).
- **SMBus** (Optional): System Management bus (SMBus) can be implemented between PCI, PCI-X, and PCI Express components on the platform and the add-in cards. Defined by the SMBUS 2.0 specification. It is a low power and lower bandwidth serial bus. The bus consists of SMBCLK (input to the add-in card) and **SMDAT** (input/output). It is a general purpose bus used for purposes that range from platform management to inputs for equipment.
- **LED\_WPAN, LED\_WLAN#, LED\_WWAN#** (Output from add-in card): Required when the associated function is implemented on the Mini PCI Express Card. These signal lines are output from the Mini PCI Express Card to drive LED indicators on the mobile platform.
- **WAKE#** (Output from add-in card): Required for add-in cards and platforms that implement the Wakeup protocol. This signal line is used in conjunction with Vaux as part of the Wakeup Protocol.
- **Power Pins** (Input to add-in card): +1.5, +3.3, and 3.3 Vaux volts. 3.3 Vaux is also known simply as Vaux.
- **Reserved**: These signal lines can only be redefined by the PCI-SIG. Neither platforms nor add-in cards can connect anything to these signal lines.
- **GND**: These signal lines are connected directly to the ground plane of the platform.

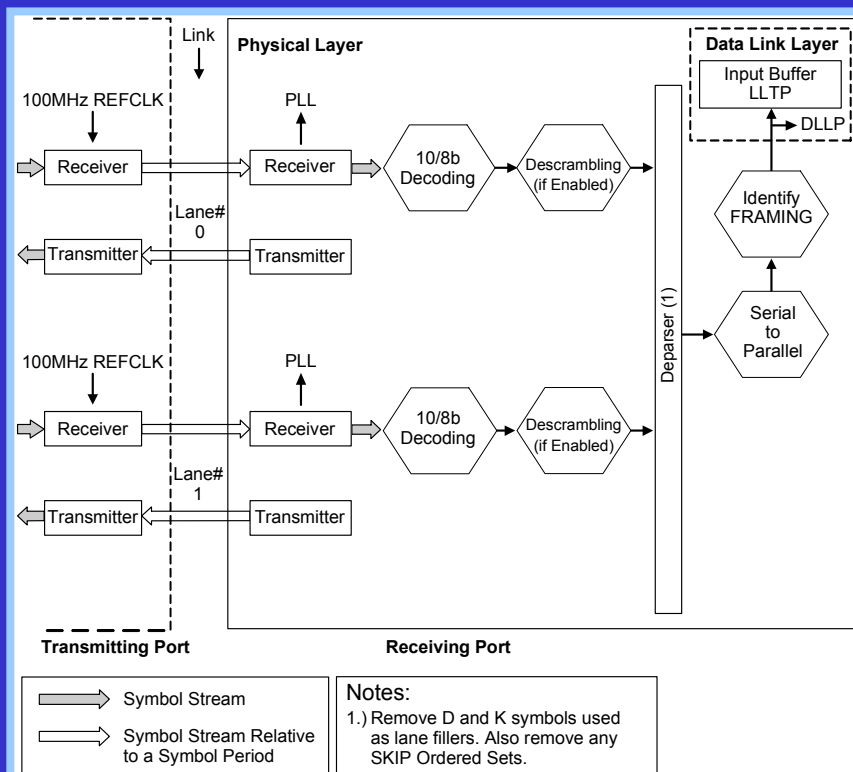
### Electrical ... Introduction

- Earlier tutorials and the Book discuss the power requirements for PCI devices and add-in cards.
- Another element of the electrical specification are the extensive details of electrical signaling. Some of the elements of electrical signaling are based process used for the components. The details of electrical signaling are left to the *PCI Express Card Electromechanical* and *Mini PCI Express Card Electromechanical* specifications and the suppliers of the components.
- The physical implementation of the circuitry at the transmitting and receiving consist of several components which are summarized in the following slides and detailed in the Book for more details.



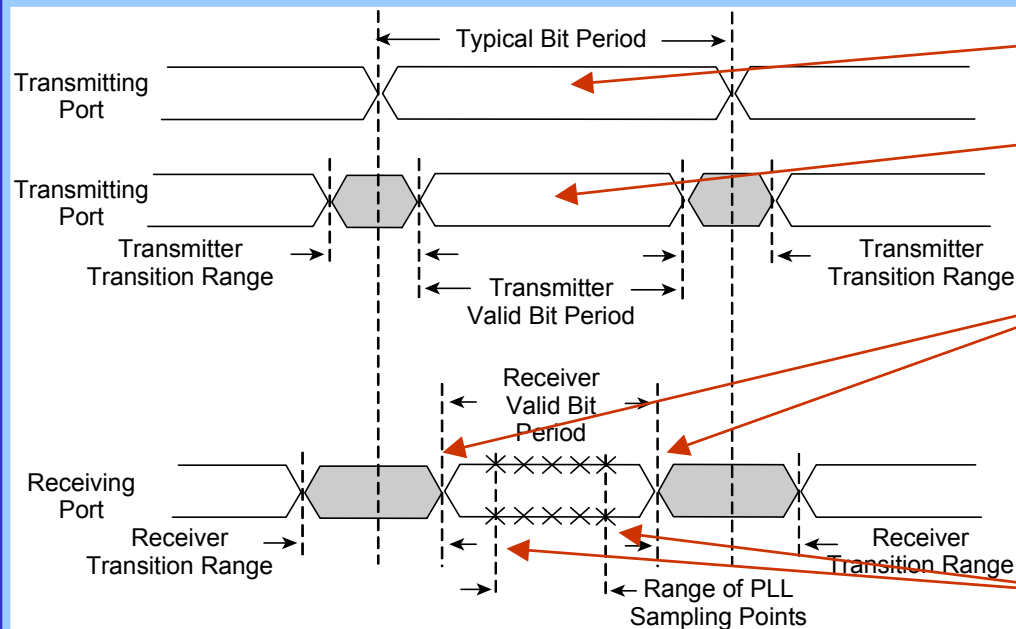
## Electrical ...Transmitting

- Once the LLTPs & DLLPs are converted to a serial stream and parsed across multiple lanes the transmitter implements the following:
  - Scrambling: Prevents the occurrence of repetitive binary patterns which cause electromagnetic interference.
  - The 8-bit format is encoded into 10-bit symbols with the integration of a reference clock.
  - The REFCLK establishes a bit period for each of the 10-bits of the symbol.



## Electrical ... Receiving

- Once the 10-bit symbols arrive, the receiver implements the following:
  - The phase lock loop (PLL) extracts the integrated reference clock to be used to provide sampling points within the bit period.
  - The 10-bit symbols are decoded into the 8-bit format used by the LLTPs & DLLPs.
  - Descrambling: Un-do the scrambling implemented by the transmitter to prevent the occurrence of repetitive binary patterns which cause electromagnetic interference.
- Once the 8-bit format is recovered the stream on each lane is deparsed into a single stream.



## Electrical ... Waveform

- Each symbol transmitted is comprised of 10 bits. Each bit defines a bit period.
- The REFCLK at the transmitting port creates the valid bit period given the tolerance of the REFCLK source and associated transmitting circuitry.
- Transmission line effects on the link reduces the width of the valid bit period at the receiver versus at the transmitter.
- The phase lock loop (PLL) at the receiving point will extract the integrated reference clock. This extracted clock is used to determine a sampling point of the symbol bit period. The tolerances of the PLL and the receiving circuitry will establish a range the sample point will exist.
- The limitation of link performance is the combination of the above considerations.

# The Complete PCI Express Reference Topic Group 6 Tutorial

Disclaimer: Intel, Research Tech Inc. and the authors of this tutorial make no warranty for the accuracy or use of the information. No direct or indirect liability is assumed and the right to change any information without notice is retained.



## Design Tools for PCI Express

The PCI Express specification is not organized by clear design topics, does not provide sufficient details to easily master PCI Express, and does not provide in depth illustrations and flowcharts to assist in designing components. Intel recognized that simply summarizing or re-wording the specification as typically done in the other design tools in the industry is insufficient ... more extensive information is are needed.

To provide designers with PCI Express design and implementation information that is easy to master and detailed enough to assist in correct design, two design tools are available:

Six Detailed Tutorials and a new and exhaustively detailed design book.

These design tools focus on Six Quick and Easy Topic Groups which simplify the mastery of PCI Express. They save a designer weeks of trying to unravel the specification and provide the assurance of correct design implementation the first time.

## Design Tools for PCI Express

### The “Book”

The primary design tool that provides total design mastery is *The Complete PCI Express Reference* book written by Edward Solari and Brad Congdon and published by Intel ... referred to as the “Book”.

The Book provides the complete and extensive narrative of detailed figures (over 250), detailed design flow charts, and exhaustive tables for the complete understanding and design assistance in over 1000 pages. The Book can be ordered at [www.amazon.com](http://www.amazon.com) ... ISBN # 0971786194.

### Detailed Tutorials

Six free Detailed Tutorials ... One self paced tutorial for each of the Six Quick and Easy Topic Groups. Each introduces PCI Express information with a narrative that complements detailed figures, flow charts, and tables for each specific Topic Group from the Book. The six free Detailed Tutorials are available at [www.intel.com/intelpress/pciexpresscomplete](http://www.intel.com/intelpress/pciexpresscomplete).

This Detailed Tutorial is of Topic Group 6  
Detailed Tutorial: *Software Considerations*  
References in the Book: *Chapters 21 to 24*

## PCI Express in Six Topic Groups

### Topic Group 1

**Tutgroup1: *Platform Architecture and Accessing of Resources within Architecture***

**References in the Book: *Chapters 1 to 4***

### Topic Group 2

**Tutgroup2 : *Packets' and Layers' Specifics and Errors***

**References in the Book: *Chapters 5 to 9***

### Topic Group 3

**Tutgroup3 : *Transaction Ordering and Flow Control Part 1 and 2 Protocols***

**References in the Book: *Chapters 10 to 12***

### Topic Group 4

**Tutgroup4 : *Power Management & Associated Protocols, Resets, Wake Events, and Link States***

**References in the Book: *Chapters 13 to 17***

### Topic Group 5

**Tutgroup5 : *Other Hardware Topics***

**References in the Book: *Chapters 18 to 21***

### Topic Group 6

**Tutgroup6 : *Software Considerations***

**References in the Book: *Chapters 22 to 24***

# Software Considerations

## Chapters 22 to 24

### Topic Group 6

A key consideration of PCI Express is the compatibility to **PCI software and the registers of the configuration address space.**

**Summary:** Of primary concern is the installed base of PCI and PCI-X compatible software. Any replacement hardware for a PCI platform must be compatible to this software and thus mimic PCI hardware. As discussed in previous slides the PCI Express platform implements the hardware with the use of virtual PCI hardware. The combination of PCI software compatibility and the use of virtual PCI hardware results in a PCI Express configuration address space that mimics PCI configuration address space.

There are three major differences between PCI and PCI Express relative to the configuration address space: One, larger configuration register block for each function in each PCI Express device. Two, the access to the configuration address space from the HOST bus segment is either via the I/O address space (PCI like) or via the memory address space. Third, additional configuration registers have been defined that are specific to new features provided by PCI Express. All of these have been reviewed in slides of other tutorials.

The number and definition of PCI Express specific configurations registers is extensive and will not be covered in detail in this tutorial. All are discussed in detail in the Book.

## Configuration Overview

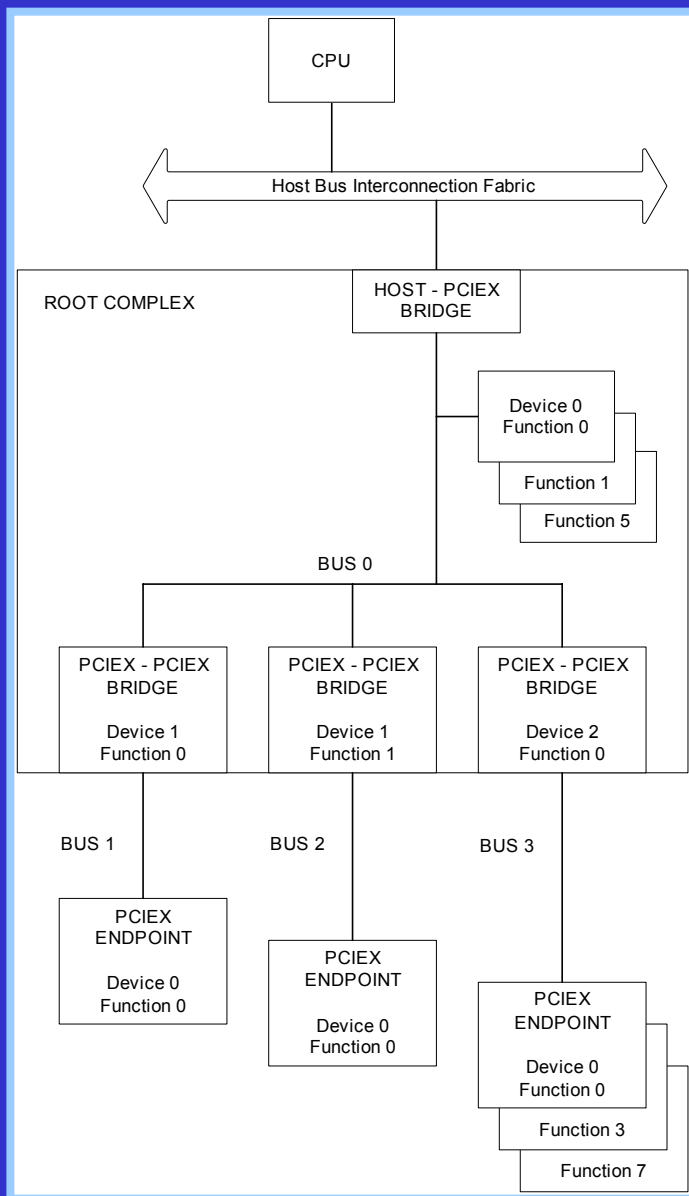
### Features of Configuration Space

- The fundamental PCI Express unit is a device. Examples of PCI Express devices are endpoints such as an embedded network controller chip or a plug-in card that operates a disk array, or bridges to remote PCI Express devices. Configuration space guarantees access to every PCI Express device in the system. Some features of configuration space include the ability to:
  - Detect PCI Express devices.
  - Identify the function(s) of each PCI Express device.
  - Discover what system resources each PCI Express device needs. System resources include memory address space, I/O address space, and interrupts.
  - Assign system resources to each PCI Express device.
  - Enable or disable the ability of the PCI Express device to respond to memory or I/O accesses.
  - Tell the PCI Express device how to respond to error conditions.
  - Program the routing of PCI Express device interrupts.

## Configuration Overview ... continued

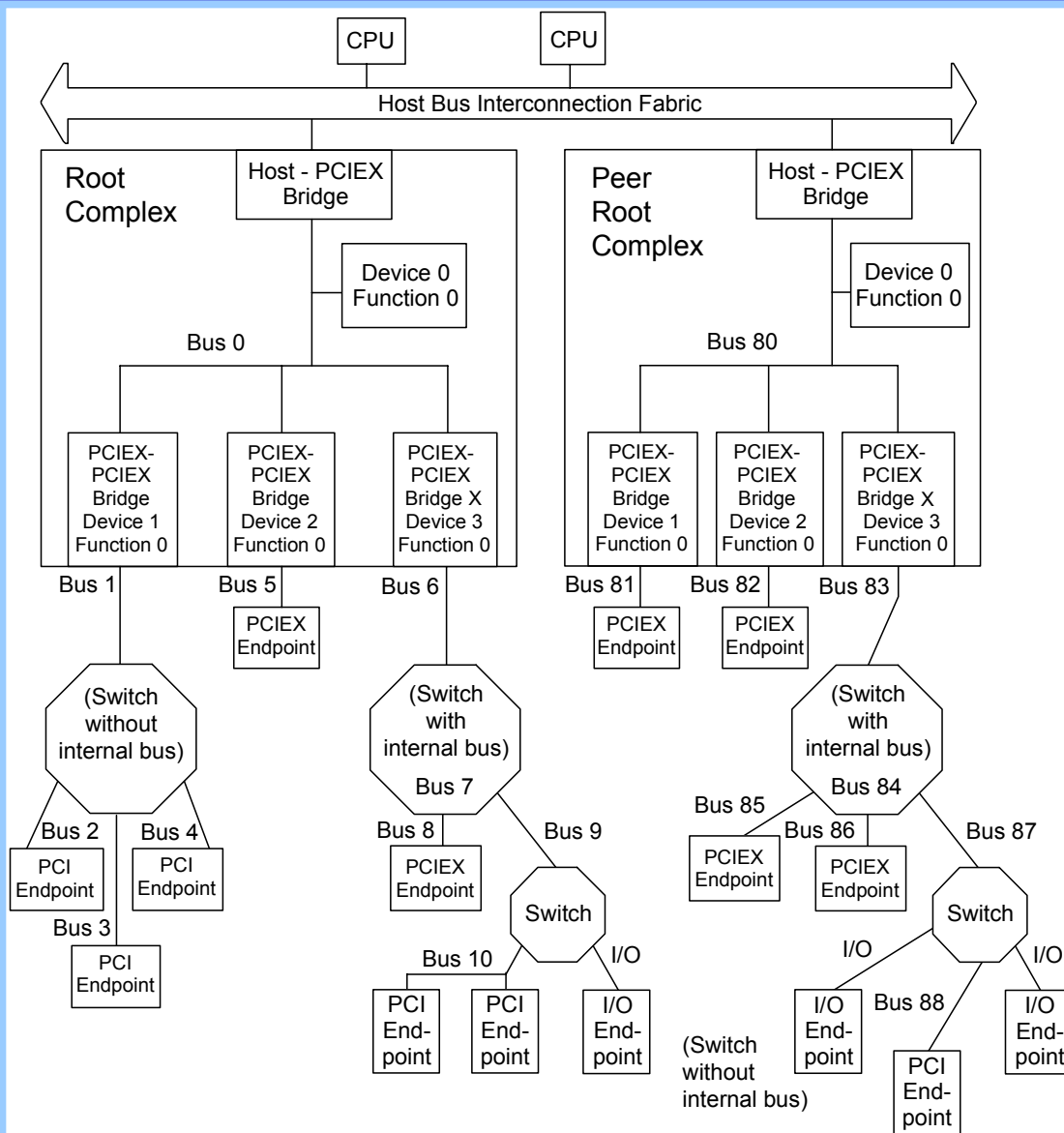
### Configuration Space Hierarchy

- In a PCI Express platform, the bus number, device number, and function number make up the 16-bit address for each unique function in a PCI Express fabric
- At least one bus always exists: bus 0. A PCI Express fabric may have as many as 256 buses (numbered 0 to 255).
- Buses hold devices, and each bus always has at least one device: device number zero. Some examples of devices are a network controller, a disk controller, or a bridge to subordinate PCI Express buses. Each bus holds up to 32 devices, numbered 0 through 31.
- In each PCI Express hierarchy, each device on a bus holds up to eight functions. For each device, function number zero is required, and functions 1 through 7 are optional. Each function is allocated 4096 bytes, separate and unique from every other function.



### Example: Small PCI Express Fabric

- Each hierarchy of the PCI Express fabric starts at Root Complex. PCI Express Bridge Devices in the Root Complex connect directly to PCI Express endpoints.
- Example enumeration shown for buses, devices, and functions. Bus 0 always resides in the primary Root Complex.
- No PCI Express bridge components or switch components are required.
- Bus numbers are assigned by software during PCI enumeration.



### Example: Large PCI Express Fabric

- Each hierarchy starts at Root Complex. PCI Express Bridge Devices in the Root Complex connect to PCI Express endpoints, bridges, and switches.
- Example enumeration shown for buses, devices, and functions. Bus 0 always resides in the primary root complex.
- Switch components may include a mix of endpoints and bridges.
- Switch components with PCI Express downstream buses include an internal bridge. Switch components without PCI Express downstream buses may or may not include an internal bus.



## Access Rules for Configuration Space

The objective of a PCI Express configuration access is to permit system software access to any specific device in the PCI Express fabric. Software can read or write any of the 256 register bytes in a PCI Express legacy device function, or any of the 4096 bytes in a PCI Express native device function. But some locations in the register map have reserved or fixed values, and other locations may not be implemented by a particular function. Nevertheless, in all cases the following access rules apply.

### Access Rules for Configuration Space Reads

- A PCI Express device is required to return data when its configuration space is read.
- A PCI Express component is required to return data when it receives a read to a bus, device, or function that does not exist. The data must be FF's with an "unsupported request completion status."
- A PCI Express device must reply with a normal completion when an un-used or reserved register is read.
- A PCI Express device must reply with a zero for any bit that is unused or reserved, unless the specification explicitly says otherwise.
- A PCI Express device must return the data value that the device is actually using.
- Software must consider the value of reserved bits as undefined.

### **Access Rules for Configuration Space ... continued**

#### Access Rules for Configuration Space Writes

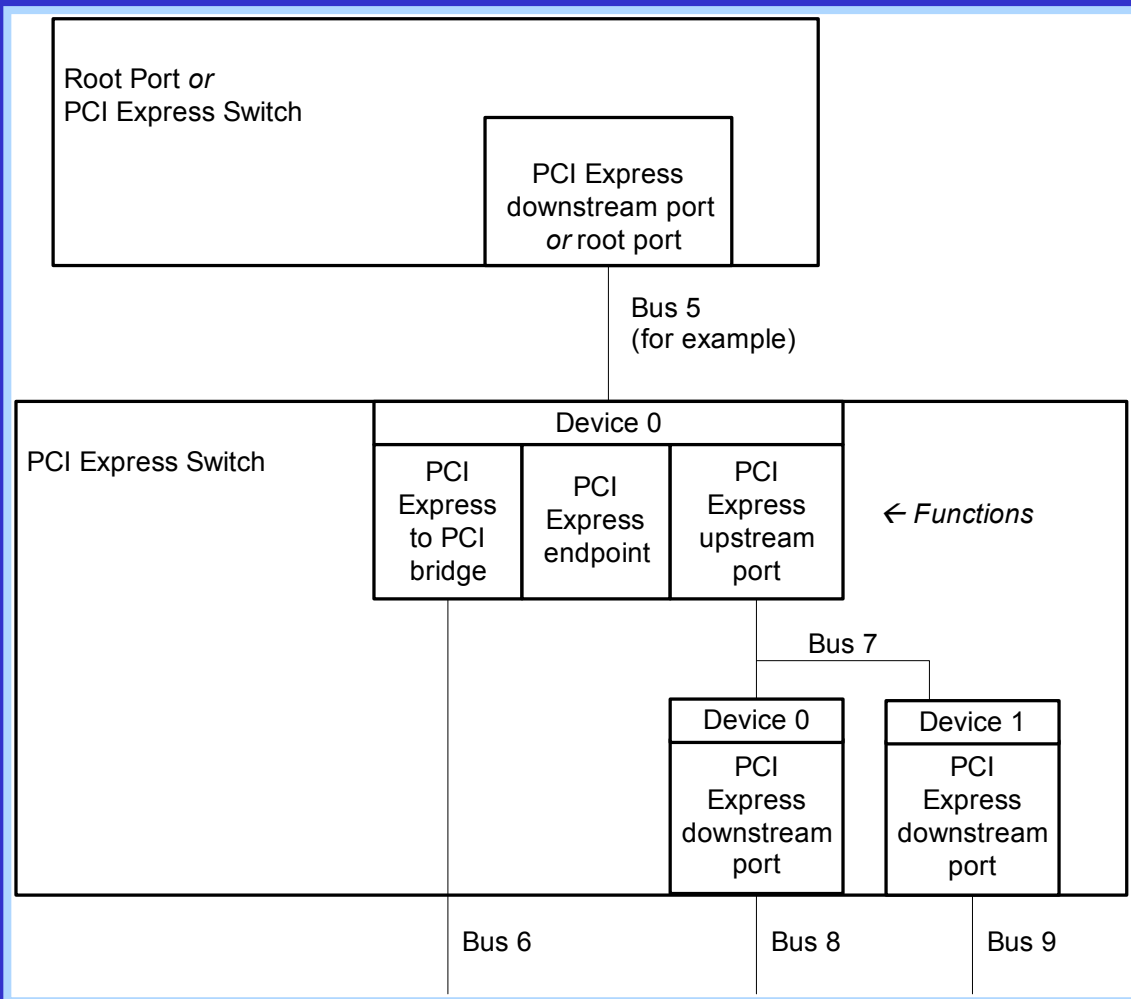
- A PCI Express device is required to ignore data during writes to reserved registers. The data is discarded, the write is changed to a no-op, and the write completes normally.
- Software must preserve the value of reserved bits when writing configuration registers. To do this, software must use a read-modify-write technique when writing to registers that hold reserved bits.

### PCI Express Bus Enumeration

- After reset, the host CPU can address bus number 0 in the Root Complex and the devices attached to it, but cannot address any other buses or devices. Every bus except for bus 0 needs its bus number assigned, which occurs after reset when system software performs PCI bus enumeration.
- During enumeration, every bus is assigned a number. Configuration registers in each device that controls a bus contains the bus number. Following is a typical algorithm by which system software assigns bus numbers.
  1. After reset, every bus is listed as zero. No device will respond to configuration accesses to any other bus number. The Root Complex is first to see configuration accesses to bus 0 and claims them.
  2. System configuration software identifies each device on bus 0 and discovers whether it includes any functions that are bridges to another bus. Bridge functions have a hardwired value of 06h in the Class Code Register. The devices can be searched in any order.
  3. When a bridge function is found, the system configuration software writes to its Primary Bus Number and Secondary Bus Number registers. For the first bridge function below the host bridge (call it bridge 2) found during enumeration, the Primary Bus Number is 0 and the Secondary Bus Number is 1.

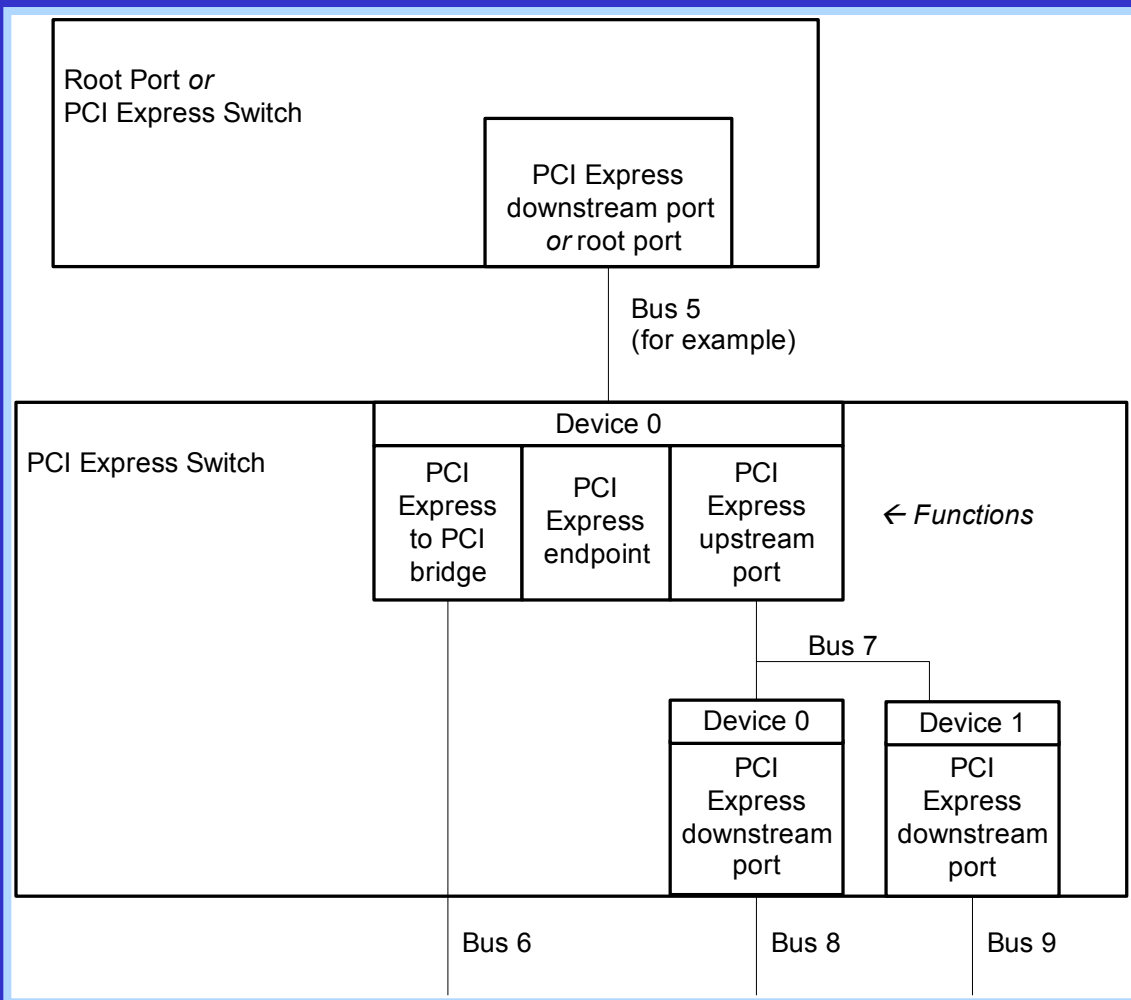
### PCI Express Bus Enumeration ... continued

- During enumeration, every bus is assigned a number. Configuration registers in each device that controls a bus contains the bus number. Following is a typical algorithm by which system software assigns bus numbers ... continued
4. Before configuration software looks for another bridge on bus 0, it looks for subordinate bridges behind bus 1. It temporarily puts the value FFh in bridge 1's Subordinate Bus Number register, then checks for bridges on bus 1.
  5. If configuration software finds a bridge on bus 1, it writes to that bridge's Primary Bus Number and Secondary Bus Number registers.
  6. Before configuration software looks for another bridge on bus 1, it looks for subordinate bridges behind bus 2. It temporarily puts the value FFh in that bridge's Subordinate Bus Number register, then checks for bridges on bus 2.
  7. Now configuration software backtracks to bus 1 and looks for more bridges on that bus. If it finds another bridge it writes to its Primary Bus Number and Secondary Bus Number registers. For example, the Primary Bus Number is 1 and the Secondary Bus Number is 3. Now configuration software puts a temporary value of FFh in this bridge's Subordinate Bus Number register and looks for bridges on bus 3.
  8. This recursive algorithm is followed until configuration software completely programs all the bridge functions with values in their Primary Bus Number, Secondary Bus Number, and Subordinate Bus Number registers.



## PCI Express Bridge Architecture

- For designers familiar with legacy PCI bridges, PCI Express bridges may at first appear confusing. In fact, no single "PCI Express bridge" function exists.
- A PCI Express bridge is either a root port bridge, an upstream port bridge, or a downstream port bridge. While a legacy PCI bridge is identified as such in the Class Code register, a PCI Express bridge (whichever type) is identified in the PCI Express Capabilities Register.



## PCI Express Bridge Architecture ... continued

- The way in which PCI Express bridges connect to each other and to other devices has certain limitations. Specifically, the following two rules apply:
  - All PCI Express devices (all devices that have functions other than root ports or downstream ports) must connect to either a root port or a down-stream port.
  - Upstream ports connect on the south only to downstream ports; down-stream ports connect on the north only to upstream ports (where north is closer to the Root Complex, south is further from the Root Complex).

### Other Rules for PCI Express Bridges

A PCI Express to PCI Express bridge in a switch consists of an upstream port function and a downstream port function. For these bridges, the following three rules apply.

- Read requests must pass un-split through the bridge.

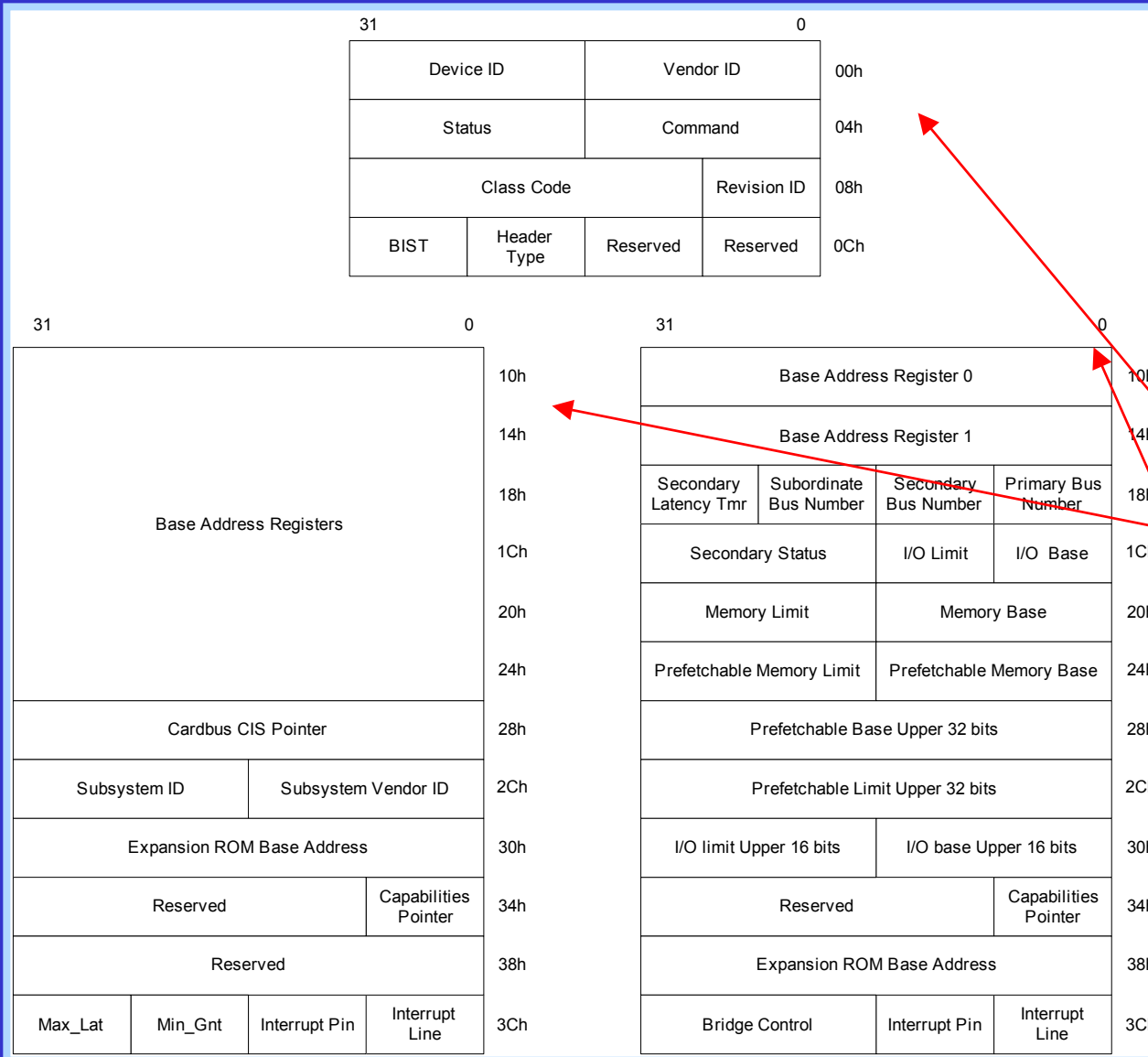
This rule means that a 4-kilobyte read request, for example, cannot be split into two or more read requests in the bridge. It is the responsibility of the destination of the read (for example, the root port) to split read requests if necessary. This rule also means that the Max\_Read\_Request\_Size register in the upstream and downstream ports should never be set to a value smaller than the value set in the Max\_Read\_Request\_Size register of any devices that uses the bridge.

- Completions and writes must pass un-split through the bridge.

This rule means that a header with a data payload cannot be split into two or more headers with smaller payloads in the bridge. This rule also means that the Max\_Payload\_Size register in the upstream and downstream ports should never be set to a value smaller than the value set in the Max\_Payload\_Size register of any devices that use the bridge.

- ECRC must pass unchanged through the bridge.

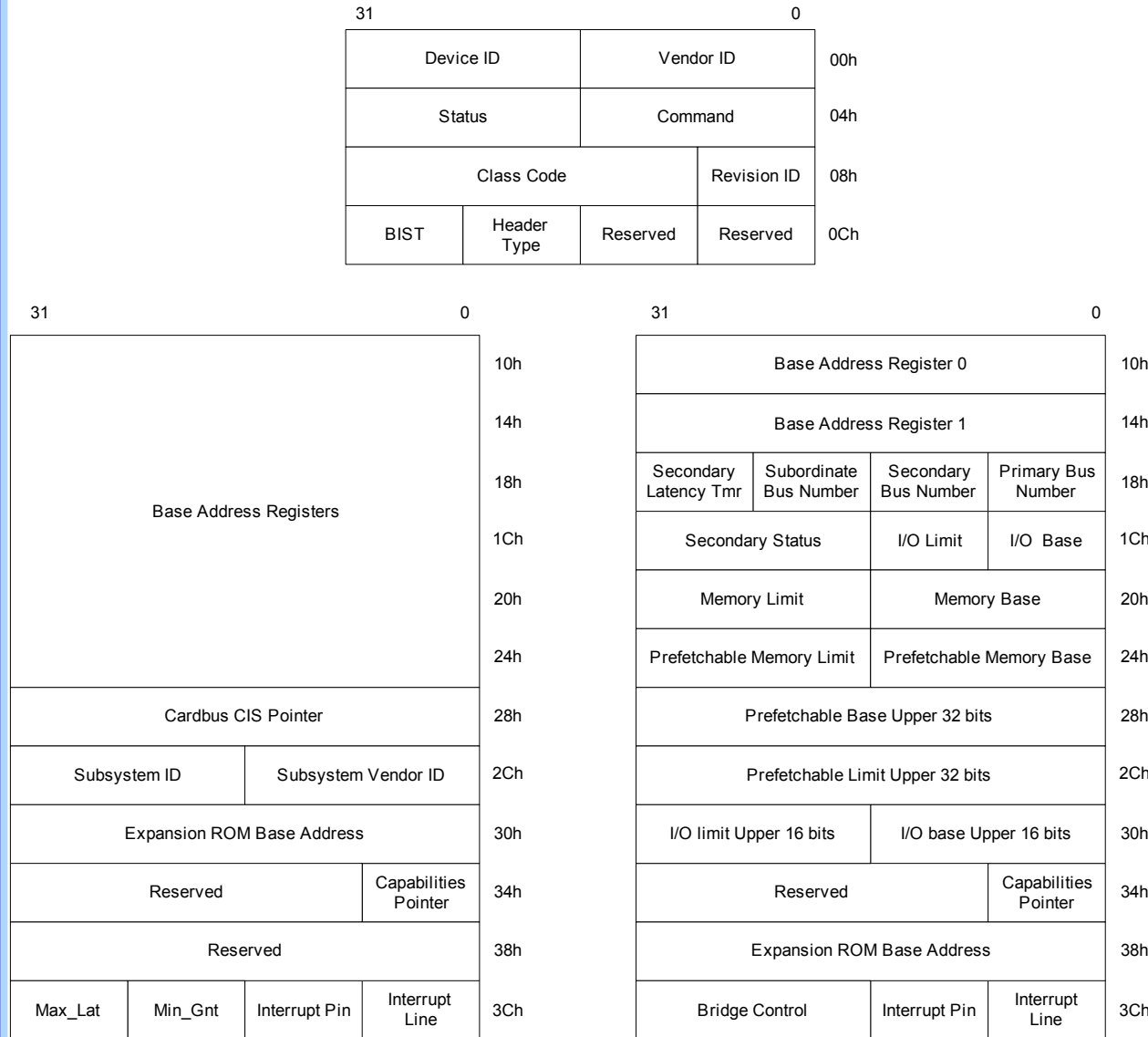
There should be no need to change ECRC if the other rules are followed.



## Configuration Registers

- Common header region.
- Type 0 header region.
- Type 1 header region.





- Device dependent region.

This is the region of PCI legacy compatible registers specific to each function, residing at configuration addresses 40h to FFh (192 bytes) in each function

- Extended configuration region.

This is the region of PCI Express registers available to each function, residing at configuration addresses 100h to FFFh (3,840 bytes) in each function.